Convio Open API Reference

Contents

Overview of Convio Open APIs....7
Getting Started....9
Configuring your Site for Convio Open APIs....9
User's Guide.... 12
General API Information....12
Servlets....12
Common Parameters....12
Using Redirect URLs....14
HTTP Status Codes....15
XML Schema Definition....16
Error Codes....16
Logging API Calls....20
Client and Server APIs....20
Client versus Server APIs....20
API Administrator Accounts for Server API Access....21
IP White List Security....21
Server Authentication using Convio APIs....21
Cross-Domain Access....22
Cross-domain Access....22
JavaScript/AJAX Programming....23
Adobe Flash Cross-domain Policy....24
Single Sign-On....25
Single Sign On (Convio as client)....25
Single Sign On (Convio as master)....25
Authentication Tokens....26
API Reference....28
Address Book API....28
addAddressBookContact....28
addAddressBookContacts....29
addAddressBookGroup....31
addContactsToGroup....32
addressBookImportOAuthCallback....33
deleteAddressBookContacts....34
deleteAddressBookGroups....35
getAddressBookContacts....36
getAddressBookGroups....38
getAddressBookImportContacts....39
getAddressBookImportJobStatus....41
importAddressBookContacts....42
parseCsvContacts....44
removeContactFromGroup....46
startOnlineAddressBookImport....47
updateAddressBookContact....49
updateAddressBookGroup....50
Advocacy API....51
getAdvocacyAlert....52
getAdvocacyAlerts....65
getAdvocacyInteractions....76
takeAction....88
Constituent API....99

create ... 99
createOrUpdate ... 103
getGroupMembers ... 109
getInteraction ... 111
getUser ... 113
getUserCenters ... 115
getUserGroups ... 117
getUserInteractions ... 119
getUserInterests ... 124
getUserTransactions ... 126
isEmailValid ... 130
isUserInterested ... 131
isUserNameAvailable ... 133
listCenters ... 134
listInteractions ... 136
listInterests ... 141
listUserFieldChoices ... 143
listUserFields ... 144
logInteraction ... 147
update ... 150
updateInteraction ... 155
Content API ... 157
listLinkProviders ... 157
listLinks ... 159
listImages ... 163
listSupportedLocales ... 166
DataSync Cross-Reference API ... 168
addXref ... 169
listComObjectsForXrefType ... 171
listForeignObjectsForXrefField ... 173
listXrefTypes ... 176
listXrefs ... 177
removeXref ... 179
Donation API ... 180
addOfflineDonation ... 181
calculateNextPaymentDate ... 191
cancelRecurringGift ... 193
donate ... 194
getDesignationTypes ... 212
getDesignees ... 213
getDonationFormInfo ... 214
getRecurringConfiguration ... 217
getRecurringGiftDetails ... 219
getRecurringGifts ... 220
getRecurringHistory ... 222
recordRecurringTransaction ... 223
refundOfflineDonation ... 225
refundTransaction ... 228
startDonation ... 233
skipNextPayment ... 246
updateRecurringBankAccountInfo ... 247
updateRecurringCreditCardInfo ... 251
updateRecurringGift ... 255
Group API ... 257
getGroupInfo ... 257
getGroupMembers ... 259

listGroups.... 261
Calendar Event API....263
getChildEvents.... 264
getDayEvents.... 265
getEvent....267
getMonthEvents.... 269
Single Sign-on API....270
authenticateUser.... 271
changePassword.... 273
getSingleSignOnToken.... 275
login....276
loginTest....281
logout....282
singleSignOn.... 283
Survey API.... 284
getSurvey....285
listSurveys.... 302
submitSurvey....306
TeamRaiser API.... 313
acknowledgeGifts....314
addDraft....315
addGift....317
addTeamraiserData....326
addTeamraiserEventManager....331
deleteDraft....332
deleteGift....333
deleteSentMessage.... 334
deleteTeamraiserAddressBookContacts....335
getCampaignByNameData....336
getCaptainsMessage.... 338
getCompaniesByInfo....338
getCompanyDonationByDay.... 343
getCompanyList.... 344
getCompanyPageInfo.... 345
getCompanyPagePreview....346
getCompanyPhoto.... 348
getCompanyShortcut....349
getCompanyTeams....350
getContactActivity.... 352
getDraft....354
getDrafts.... 355
getEventDataParameter....357
getFlowStep....360
getFundraisingResults.... 361
getGift....362
getGiftCategories.... 363
getGifts.... 364
getMessageLayouts.... 367
getNewsFeeds....368
getOrganizationMessage.... 370
getParticipantCenterWrapper.... 371
getParticipantFBConnectInfo....372
getParticipantProgress....373
getParticipants.... 374
getParticipationType.... 379
getParticipationTypes....380

getPersonalDonationByDay....384
getPersonalMediaLayout....385
getPersonalPageInfo....386
getPersonalPagePreview....387
getPersonalPhotos....389
getPersonalVideoUrl....390
getPublicEventTypes....391
getRecentActivity....392
getRegisteredTeamraisers....394
getRegistration....398
getRegistrationDocument....400
getSentMessage....404
getSentMessages....405
getShortcut....407
getSuggestedMessage....408
getSuggestedMessages....410
getSurveyResponses....411
getTeam....413
getTeamCaptains....414
getTeamDivisions....416
getTeamDonationByDay....416
getTeamGifts....417
getTeamMembers....420
getTeamPageInfo....421
getTeamPagePreview....422
getTeamPhoto....424
getTeamRoster....425
getTeamsByInfo....427
getTeamShortcut....434
getTeamraiserAddressBookContact....435
getTeamraiserAddressBookContacts....436
getTeamraiserAddressBookContactsByIds....440
getTeamraiserAddressBookContactsGiftHistory....441
getTeamraiserAddressBookFilters....442
getTeamraiserAddressBookGroupContacts....445
getTeamraiserConfig....446
getTeamraiserEventManagerRoles....448
getTeamraiserResultsData....450
getTeamraiserSuggestion....451
getTeamraisersByDistance....452
getTeamraisersByInfo....458
getTentingSearch....464
getTentmate....467
getTopCompaniesData....468
getTopDonors....469
getTopParticipantsData....471
getTopTeamDonors....473
getTopTeamsData....475
getUpsell....476
joinTeam....478
leaveTeam....479
listTeamraiserEventManagers....480
previewMessage....481
processRegistration....482
removeCompanyPhoto....485
removePersonalPhoto....487

removeTeamPhoto..... 488
removeTeamraiserEventManager..... 489
sendTafMessage..... 490
setTeamCaptains..... 491
updateCaptainsMessage..... 493
updateCompanyPageInfo..... 494
updateCompanyShortcut..... 497
updateDraft..... 498
updatePersonalMediaLayout..... 500
updatePersonalPageInfo..... 501
updatePersonalPagePrivacy..... 503
updatePersonalVideoUrl..... 504
updateRegistration..... 505
updateShortcut..... 507
updateSurveyResponses..... 508
updateTeamInformation..... 510
updateTeamPageInfo..... 511
updateTeamShortcut..... 512
updateTeamraiserAddressBookContact..... 513
updateTeamraiserData..... 515
updateTentingStatus..... 517
uploadCompanyPhoto..... 518
uploadPersonalPhoto..... 519
uploadTeamPhoto..... 521

Overview of Convio Open APIs

While the Convio platform offers extensive flexibility to serve the online marketing needs of non-profit organizations, clients often need separate, custom-built applications that integrate with the Convio solution. Convio APIs provide open access to constituent and campaign data, allowing you to innovate in ways that engage more people. Using Convio APIs, you can develop custom applications that leverage Convio as a data management and transaction processing platform.

Address Book API

The Address Book API provides access and management functions for TeamRaiser Address Books.

Advocacy API

The Advocacy API provides methods to query and report on Advocacy alerts and user interactions.

Calendar Events API

The Calendar Events API provides query access to Convio Calendar Events (TeamRaiser events are exposed separately through the TeamRaiser API).

Constituent Management API

The Constituent Management API supports multiple methods to create, update, and query user data in the Convio Constituent360 database.

Donation Processing API

The Donation Processing API allows you to extend and customize the online giving and payment processing capabilities of the Convio system.

Single Sign-On API

The Single Sign-On API provides methods to authenticate users and securely establish a logged-in session with the Convio server using API calls. These API methods support both Convio-as-master and Convio-as-client implementations allowing you to use either Convio or an external system as the master authentication service.

TeamRaiser API

The TeamRaiser API gives you the means to customize and extend the TeamRaiser application, the Participant Center and the web pages for TeamRaiser teams, companies, and participants. This API includes methods to:

- record, query, manage, and search TeamRaiser Events,
- manage TeamRaiser Teams,
- manage Team, Company, and Personal web pages,
- record, query and manage Gifts,
- query and update Surveys,
- create, send, query, and manage messages to TeamRaiser Participants and Contacts.

Survey API

The Survey API gives you the means to customize your constituent's interactions with Convio surveys by providing total control over the look and feel of the survey. By implementing the display of the survey's questions you can customize the behavior of questions to make the answer a user provides to one question change the behavior of another question, such as: showing or hiding other questions, changing the available answers in other questions, or even making certain questions required (or not required) by implementing you own script-based validation. This API includes methods to:

- list the surveys a user has access to view,
- retrieve the details and questions for a single survey,
- submit responses to a single survey,

Getting Started

Configuring your Site for Convio Open APIs

Overview

Before you can use Convio Open APIs, you must configure your Convio web site to enable and control API access. The Open API Configuration tool allows Convio site administrators to configure your site for API access. The Open API Configuration tool is also used to manage cross-domain access to the Open APIs from JavaScript and Flash clients, and allows logging of Open API calls and downloading/viewing of log files.

Convio Open APIs permit clients to integrate features and functionality of the Convio platform into other web server applications or web page content. These APIs can be divided into two different categories based on how they are accessed. Each offers slightly different features and has different configuration requirements. These categories are:

- **Client APIs:** APIs intended to be accessed directly from a constituent's web browser using client-side scripting. These APIs are the easiest to use, requiring no server-side programming. They may or may not require authentication or session security, depending on the information they expose.
- **Server APIs:** APIs for use by other application servers. Some may be used as server-side proxies for the Client APIs described above. These APIs respond only to secure requests from trusted IP addresses, and require valid API Administrator credentials with each request. As such, they are able to expose some administrative functions.

The Open API Configuration tool allows a site administrator to easily configure access for both categories of Convio Open APIs, and also for *Convio Web Services*.

Accessing the Open API Configuration Tool

To access the Open API Configuration tool:

1. Log in to Convio as an administrator.
2. Go to **Setup**->**Site Options**.
3. Click the **Open API Configuration** tab.

A dashboard screen will open showing the configuration status of the various Convio API options. Some of these may be configured automatically by the Open API Configuration tool. Configured options appear in green next to a check mark, while options that are not yet configured appear in red beside an "X." **Note**: Not all options must be configured for all API access methods.

The sections of the dashboard correspond generally to different access methods. Edit the settings for each section by clicking the link next to the section heading. The sections available are:

- **Configure API Keys** -- Use this section to configure the site for either client-side or server-side API access. Note: if you wish to use server-side APIs, you must also define the options under the "Configure API to allow server access" section.
- **Configure API to allow server access** -- Use this section to list the addresses of servers authorized to access the Convio Open APIs (Server), and to create API Administrator accounts to control access via Convio Open APIs (Server) and Convio Web Services.
- **Configure JavaScript/Flash access to Open APIs** -- Use this section to control cross-domain access from web pages containing the Convio Open API JavaScript class library, or from Flash applications that reference Convio's cross-domain policy files (crossdomain.xml).
- **Configure AJAX proxy service** -- Use this section to designate external domains to which Convio may proxy AJAX JavaScript requests.
- **Configure Web Services** -- Use this section to configure the Convio Web Services SOAP-protocol APIs. For additional information on Convio Web Services, see the *Convio Web Services website*.

Click the corresponding **Edit** action link on the main Open API Configuration page to access and configure your site for use with the APIs.

Open API Keys

These API keys must be configured to allow access through any of the REST-style Convio Open APIs (Client) or Convio Open APIs (Server). API Key is the only required field, it may be entered or changed to a different value here. The other available options are:

- **Convio API Key --** This key is an arbitrary value that must be passed as the value of the **api_key** parameter when invoking Convio Open APIs either from a server script or web form. This is not a secure password and it may be included in a hidden field of a web form.
- **Convio API Secret Key --** This key is an arbitrary value that will be used by Convio APIs to sign responses. This key should never be passed in API requests directly and should be a reasonably complex string. It is used by the Convio server along with additional data to generate a hash signature on redirect responses to API calls when the sign_redirects parameter is set. For additional information on its use, see *Using Redirect URLs*
- **Generate a new Secret Key --** If the value of your secret key has been compromised, click this button to generate a new one.
- **Digital signature encryption algorithm --** Determines the encryption method when the option to **sign_redirects** is used with API calls. The default is MD5, but SHA-1 offers a more secure encryption algorithm.
- **Single Sign-On APIs** -- Check this box to enable *singleSignOn*, which allow a trusted external host to log in a Convio constituent using a token returned by *getSingleSignOnToken*. Clearing this check box disables the `singleSignOn` method of the SSO API. For additional information see *Single Sign On (Convio as client)*.

Open API Server API Configuration

This screen allows site administrators to designate which external systems have access to Server APIs by adding them to the **Convio API IP White List**.

Note: The IP White List only permits access to the Convio Open API (Server) REST-style interfaces. Convio Web Services SOAP APIs use a different IP White List, and Open API (Client) interfaces allow connections from any IP.

The available options are:

- **Add New IP Range to List** -- Specify the IP Range for servers authorized to access the Open API (Server) interfaces using *CIDR* notation.
- **Add Current IP Address to list** -- This button automatically adds the IP address of the system on which you are running the Convio Admin application to the White List. This can be useful for testing.
- **Remove Selected IP Range** -- This button removes selected server IP ranges from the White list.
- **API Administrative Accounts** -- Set up accounts used by both the Open API (Server) REST-style APIs and the Convio Web Services SOAP APIs. API Administrator Account credentials must be passed in the login_name and login_password parameters on Server API calls; they are not used by Client API calls. For more information see *API Administrator Accounts for Server API Access*.

JavaScript and Flash Configuration

Use this page to control cross-domain access from other domains that host Web pages containing the Convio Open API JavaScript class library. These settings also determine how Flash applications hosted on external domains may access the APIs using Convio's cross-domain policy files (crossdomain.xml). You can grant different levels of access (basic or trusted) to different domains.

Options on this page are:

- **Allow JavaScript/Flash API from these domains --** List any external domains which should have basic access. These domains will be able to create constituents (but not update) and retrieve public data. The list should be comma separated and can include wildcards (e.g. *.convio.com).
- **Trust JavaScript/Flash API from these domains --** List any external domains which should have trusted access. Scripts hosted from these domains will be able to retrieve an authorization token for use with the Convio APIs. Trusted domains will be able to retrieve personal information about constituents and update constituent data.
- **Minify JavaScript files** -- Set to TRUE (recommended and default) to minify the Convio API JavaScript files when they are generated. The minified versions will load faster and improve overall user experience. The only reason to not minify the JavaScript files is so that you can run the JavaScript using a debugger.

- **Require Flash files to be served securely** -- Set to TRUE (recommended and default) if any Flash movie (.swf file) that is used to access the APIs must be retrieved from a secure URL. Note that this is the URL for the .swf file itself, not the page on which it is hosted.

These settings apply only to JavaScript or Flash access to Convio Open APIs (Client). They do not apply Server APIs or to Convio Web Services.

AJAX Proxy Configuration

Use this page to designate other domains from which services may be proxied through the Convio web server.

Options on this page are:

- **AJAX Proxy Allowed Domains** -- Use this field to specify a list of domains to which Convio may proxy AJAX requests.
- **Forward cookies --** Set this flag to forward cookies to proxied services (note restrictions).
- **Pass arguments through template rendering --** Set this flag to accept Convio template expressions as tags in script requests from clients and render them before passing the result to the proxied service.
- **Maximum input body length in MBytes** -- Use this value to limit the size of the request that Convio will pass to the proxied service.

Configuring or Viewing API Logs

Use the **Configure or View API logs** page, located under "Related Actions," to enable various levels of logging for debugging and trouble-shooting purposes and to view the generated log files. One file is maintained for each instance of the Convio web server running for any given Convio Website (one for each JVM).

Convio API Logging Level -- Controls what requests and responses are logged:

- OFF (no logging)
- ERROR (log only errors)
- INFO (log request parameters and responses)
- DEBUG (most detailed logging)

For performance reasons, INFO and DEBUG logging are typically used only in development and debugging, whereas ERROR logging or OFF may be used in day-to-day operation. Log file size is limited to a 1MB file with one backup (2MB total log size per site, per JVM), to guard against a potential out-of-memory condition in the event that logging is accidentally left on.

User's Guide

General API Information

Servlets

Servlet URL

The Convio Open APIs are ReST-style APIs, invoked by passing form data in an HTML POST or GET request to the servlet URL of the API.

In most cases `secure2.convio.net` or `secure3.convio.net` matches a client's secure domain. However, in some cases a client will have their own branded secure domain and this will be different. If your organization uses Convio for fundraising, the correct secure domain for the API is identical to the secure domain for your fundraising forms. So if the sample above does not work, look for the unique secure domain used for your organization and replace the `secure2.convio.net` portion above with this URL.

Here, `organization` is a client's unique identifier in the Convio system. Every client has a unique value that represents their instance of the product. If your organization uses a custom secure URL for donations, the `organization` portion of the URL may not be necessary, contact support for assistance.

The `site` part of the URL indicates which channel ("site" versus "admin") the API is to use to authenticate a logged-in user's session. Generally you will use the "site" channel, however, if you are extending the Convio Admin user interface via a client-side API, you should use "admin" instead, since Admin users log in via this channel. The following servlets support the use of "admin" channel URLs: CRAdvocacyAPI, CRConsAPI, CRContentAPI, CRGroupAPI, and CROrgEventAPI.

Common Parameters

Common Client and Server API Parameters

The following parameters are supported by all Convio Open APIs, both Client APIs and Server APIs:

method

Required. The method to be invoked. Method names are case-sensitive, and by convention begin with a lower-case letter with camel-cased words.

Type xsd:string

v

Required. The version number for the API. Currently only version `1.0` is supported.

Type xsd:string

api_key

Required. An arbitrary value that must be passed when invoking the Convio Client and Server APIs. The value passed by the caller must match the value in the CONVIO_API_KEY site configuration parameter, which is unique for each Convio site. This value will be the same for all API methods. Note that this value is not considered secure and may be visible to end users.

Type xsd:string

response_format

Optional. The desired response format. Either "xml" or "json" may be specified. If no value is specified, then "xml" is the default.

Type xsd:string

center_id

Optional. The ID of a center (in a multi-center enabled site) to associate with the current session.

Type xsd:nonNegativeInteger

source

Optional. Text to associate with newly created constituent records to identify how or from where the record was added. This is recognized by all API methods even though most of them do not directly create constituent records. This is because the value is remembered in the visitor's current web browser session and used if they take any action, either via API call or browser interaction, that causes a new constituent record to be created.

Type xsd:string

sub_source

Optional. Further detail to save with the "source" information.

Type xsd:string

Common Client API Parameters

The following parameters are supported by Convio Client APIs:

error_redirect

Optional. Specifies a URL redirect to send back to the browser after processing if an error has occurred.

Type xsd:anyURI

redirect

Optional. Specifies a URL redirect to send back to the browser after successful processing.

Type xsd:anyURI

success_redirect

Optional. Specifies a URL redirect to send back to the browser after successful processing.

Type xsd:anyURI

sign_redirects

Optional. Indicates that redirect URLs should include a digital signature. The digital signature consists of two additional parameters in the URL. The ts parameter will contain a timestamp associated with the completion of the API call. The sig parameter will contain an MD5 hash of the URL query string up to but not including the sig parameter, and the CONVIO_API_SECRET_KEY value.

Type xsd:boolean

suppress_response_codes

Optional. Specifies that all responses will be returned with an HTTP 200 OK status code - even errors. This parameter exists to accommodate Flash and JavaScript applications running in browsers that intercept all non-200 responses. If `true`, the client must determine error states by parsing the response body. Default is `false`.

Type xsd:boolean

Common Server API Parameters

The following parameters are supported by all Convio Server APIs:

login_name

Required for server API calls, invalid for client API calls. The user_name of the Convio administrative account that was created for API access. Note that this must be a special administrator account used just for API access.

Type xsd:string

login_password

Required for server API calls, invalid for client API calls. The password of the Convio administrative account that was created for API access. Note that this must be a special administrator account used just for API access.

Type xsd:string

Using Redirect URLs

All client APIs provide for specifying redirect URLs. As a result, AJAX is not needed to invoke the APIs. Convio APIs can redirect after method execution to a specified URL, either on the Convio Web server or on another system. The redirect, success_redirect and error_redirect parameters are optional parameters which can be passed on invocation of any client API.

Specifying a redirect parameter may result in a HTTP 302 "Redirect" status being returned to the client's browser following method invocation. The call may or may not have succeeded, depending on which parameter was used:

- success_redirect - redirects to the specified URL if the call succeeded
- error_redirect - redirects to the specified URL if the call failed
- redirect - redirects to the specified URL regardless of the result

The sign_redirects parameter is an optional boolean parameter, which, if **true**, instructs the Convio server to sign redirects with a timestamp and the Convio API secret key.

Using the sign_redirects Parameter

Using the sign_redirects option, the target can verify that a redirect request originated with Convio. Otherwise it is impossible to trust that the page being served in response after an action is completed (e.g. a successful login or a completed donation) is being requested as a byproduct of completing that action or if it was just coming from any other source (e.g. a link or direct entry into the browser address bar).

The URL is signed by adding a timestamp to the query string. Then an MD5 or SHA-1 hash of the URL query string plus the CONVIO_API_SECRET_KEY are calculated and the result is appended to the query string. The CONVIO_API_SECRET_KEY should be a value that is not easily guessed and should never be stored or communicated in clear text. This is in direct contrast to the CONVIO_API_KEY, which can be communicated in clear text and is frequently visible in URLs or in the source code view of a web page.

1. Pull that portion of the URL query string after the "?" and up to the argument "&signature=".
2. Append the secret key to that portion of the query string.
3. Generate the appropriate hash (MD5 or SHA-1) for that string.
4. Compare the hash with the value of the "signature" argument.
5. Assuming that the values match, you can be confident that the URL was generated by Convio.
6. Compare the value of the "ts" argument in the URL with the current time from your system in seconds since 1/1/1970. The values should be close enough (within a few seconds provided your system clock is accurate) to ensure that this is not a replay attack.

Automatic Redirection Via Insecure Channel

When a redirect is specified to a non-Convio domain, the API will first check to see if it has already pushed a session cookie on the insecure channel. If not, it will actually generate 2 consecutive redirects to ensure that the session cookie is passed on the insecure channel.

This is particularly important in single-sign-on implementations to ensure that links from the partner system connect back to the same session that authenticated the user originally.

Substitution of Request Params in Redirect URLs

Request parameters can be substituted into the redirect URL by specifying ${paramname} as the value of a query string parameter in the URL. This functionality is useful for redisplaying the data that was entered by the user.

For security reasons, some parameters are never passed in a redirect URL. These include:

- card_number
- cvv_number
- card_exp_month
- card_exp_year
- password
- user_password
- retype_password
- old_password
- user_name
- token
- sso_auth_token
- login_name
- login_password

Substitution of Response Elements in Redirect URLs

Any value that would normally be returned in the response XML can be substituted into the redirect URL. The response values use XPath syntax to identify the value from the response XML (for example, error_code= ${erorResponse/code}). This feature is necessary for error handling, displaying confirmation codes, and retrieving cons_id from the return. Because the response is a structured XML document, you may use XPath syntax for accessing the variables from the response.

Typically, the response documents are fairly simple and the path used to access a variable will be a simple directory path.

HTTP Status Codes

HTTP Response

The HTTP status code of 200 usually indicates a successful invocation. If an error occurs during processing, HTTP status code 403 is usually returned, and the structure of the response is an `errorResponse` object. It contains two elements: a `code` element and a `message` element. The `code` element contains a number corresponding to the error conditions documented on the error code list page. The `message` element contains a text message that may provide further detail about the error. The response is formatted as XML by default, but the response_format input parameter can be used to specify either XML or JSON formatting.

The following HTTP status codes may be returned by calls to Convio APIs.

Code	Description
200	OK. Generally, indicates a successful invocation. May also be returned on error if call includes `suppress_result_codes=true`.
302	Redirect. The invocation may have been successful or an error may have occurred. A redirect parameter was specified in the request so the response is a redirection to that URL.
403	Forbidden. The most common error status, it indicates that the request was received and understood but an error occurred. Parse the content text for more information.

405	Bad Method. Indicates that the request was not sent using the "POST" HTTP method.

XML Schema Definition

The structure of Convio API methods' XML response documents are defined in XML Schema Definition (XSD) files. The top-level XSD file is located at *http://service.convio.net/xmlschema/crm.public.v1.xsd*. Within the XML response documents of API calls, "xsi:schemaLocation" attributes reference this location. The top-level XSD file includes several other XSD files that define the structures used by each API.

href="http://service.convio.net/xmlschema/crm.public.v1.xsd	Top-level XSD that defines the root documents for API methods' responses and common error response documents. The root document elements generally reference a type definition that is included from an API-specific XSD.
http://service.convio.net/xmlschema/crm.public.common.v1.xsd	XSD that defines some common data structures used by other APIs. This file is included by several other API XSD files.
http://service.convio.net/xmlschema/crm.public.addressbook.v1.xsd	AddressBook API XSD that defines the types for elements returned by Address Book API methods.
http://service.convio.net/xmlschema/crm.public.advocacy.v1.xsd	Advocacy API XSD that defines the types for elements returned by Advocacy API methods.
http://service.convio.net/xmlschema/crm.public.constituent.v1.xsd	Constituent API XSD that defines the types for elements returned by Constituent API and Single Sign On API methods.
http://service.convio.net/xmlschema/crm.public.content.v1.xsd	Content API XSD that defines the types for elements returned by Content API methods.
http://service.convio.net/xmlschema/crm.public.datasync.v1.xsd	DataSync API XSD that defines the types for elements returned by DataSync API methods.
http://service.convio.net/xmlschema/crm.public.donation.v1.xsd http://service.convio.net/xmlschema/crm.public.recurring.v1.xsd	Donations API XSD that defines the types for elements returned by Donations API methods.
http://service.convio.net/xmlschema/crm.public.group.v1.xsd	Group API XSD that defines the types for elements returned by Group API methods.
http://service.convio.net/xmlschema/crm.public.orgevent.v1.xsd	Events API XSD that defines the types for elements returned by Events API methods.
http://service.convio.net/xmlschema/crm.public.survey.v1.xsd	Survey API XSD that defines the types for elements returned by Survey API methods.
http://service.convio.net/xmlschema/crm.public.teamraiser.v1.xsd	TeamRaiser API XSD that defines the types for elements returned by TeamRaiser API methods.

Error Codes

Generic API Error Codes

These error codes may be returned by any API call; they are not specific to a particular application.

Code	Description
1	Unable to process request. This generally indicates that an exception occurred within the internals of executing the API.

2	Incorrect API key. Verify that the value of the parameter api_key matches the value of the SDP CONVIO_API_KEY.
3	Authorization failed. Verify that the credentials presented are correct. This could also mean that the API is valid only for logged in users or if creating a new user that it is valid only if not already logged in.
4	Request not allowed from source IP address. Verify that the originating IP address is entered correctly in CIDR format (xxx.xxx.xxx.xxx/yy) in CONVIO_API_IP_FILTER
5	Method is restricted; the API requires an auth token.
6	No method specified or the method specified is invalid. Make sure that you are passing in a correct value for the argument "method" and note that the method names are case-sensitive.
7	Single SignOn failed. Either an sso_auth_token was not provided or the token provided was invalid.
8	A request that should apply to a single constituent matched more than one constituent. This occurs when an email address is not unique.
9	One or more required parameters were missing from the request.
50	The specified list criteria are invalid.

Advocacy API Error Codes

The following error codes may be returned by calls to the server-only Advocacy API (SRAdvocacyAPI).

Code	Description
5801	Issue not found.
5802	Invalid alert_status argument.
5803	Invalid alert_type argument.
5804	Invalid publish date.
5805	Invalid expire date.
5806	Constituent not found.
5807	The publish argument must not be after the expire argument.
5808	Only ACTIVE and EXPIRED alert_status values are allowed.

Single Sign-On API Error Codes

The following error codes may be returned by calls to `singleSignOn` or `login` (CRConsAPI or SRConsAPI).

Code	Description
200	Missing user_name on login request.
201	Missing password on login request.
202	Invalid/incorrect user_name or password on login request.
203	Internal error resulted in failure to email user_name and password.
204	Indicates that the user is not currently logged in.

Constituent API Error Codes

The following error codes may be returned by calls to the server or client Constituent API (CRConsAPI or SRConsAPI).

Code	Description
10	Update failed: permission error.
11	Add failed: duplicate.
12	Update failed: duplicate user name.
13	Update failed: duplicate e-mail address.
14	Method not allowed.
15	Update failed: matched multiple records.
16	Constituent record not found.
17	(Deprecated) No method was specified.
18	Attempt to access a user field did not match any records.
19	Error returning group membership.
20	Invalid password.
21	Change password error: the old password is required.
22	Validation error.

Donation API Error Codes

The following error codes may be returned by calls to the client-only Donation API (CRDonationAPI).

Code	Description
101	Donation failed.
102	Donation form not found.
103	Donation level not found.
104	Donation campaign not found.

Event API Error Codes

The following error codes may be returned by calls to the client-only Calendar Event API (CROrgEventAPI).

Code	Description
1901	Event not found.
1902	Invalid month/year.
1903	Invalid date value.

TeamRaiser API Error Codes

The following error codes may be returned by calls to the client-only TeamRaiser API (CRTeamraiserAPI).

Code	Description
2601	No event specified, or the event could not be found.
2602	Some required parameters were missing or invalid.
2603	An event registration record could not be found.
2604	You do not have permission to perform the specified action.
2605	The specified survey could not be found.
2606	The specified question could not be found.

2607	An unexpected error occurred while updating the event registration.
2608	The specified upsell could not be found.
2609	An error occurred while logging an interaction.
2610	An invalid value was provided.
2611	The requested results are currently being calculated. Please wait a minute and try again.
2612	A duplicate email address was detected.
2613	A duplicate shortcut was detected.
2614	The specified team was not found or you do not have permission to access it.
2615	The specified company was not found or you do not have permission to access it.
2616	An unexpected error occurred with the page you requested.
2617	An error occurred retrieving the requested list of data. Please wait a minute and try again.
2618	The specified team was not found.
2619	The specified event is not currently accepting offline gifts.
2620	The message ID you specified was not valid.
2621	You do not have permission to perform the requested action on the message
2622	One or more email addresses passed in are malformed.
2623	The contact for the ID you provided could not be found.
2624	The offline gift failed for the request donation.
2625	You tried to change the captains for a team, but multiple captains is not enabled for this event.
2626	You tried to make too many captains.
2627	You cannot submit a new company name and assigned a company id together.
2628	A team may not be assigned to a company.
2629	A new company cannot be assigned to the team.
2630	The team update failed for the request.
2631	The company chosen is invalid.
2632	The file could not be uploaded..
2633	The file uploaded contains an unsupported content type.
2634	The privacy of the page could not be updated.
2635	The tenting update failed.
2636	Tenting is not allowed.
2637	The invitation to tent was previously accepted.
2638	The invitation to tent was previously rejected.
2639	A tenting invitation has been sent or is awaiting a reply.
2640	No tenting invitation exists to reject.
2641	Tenting invitation has already been sent, awaiting reply, or has already been accepted.

2642	Tentmate may not be the same user as yourself.
2643	Gift payment type submitted is not allowed.
2644	You are not allowed to enter a gift.
2645	The message subject is required.
2646	The video url passed in is not valid.
2647	The content contains one or more invalid html tags.
2648	The gift contains an invalid gift category.
2649	No permission to retrieve the team roster
2650	Team field may not be updated.
2651	Team field may not be updated.
2652	The message body content is required.
2653	One or more recipients is required.
2654	The file uploaded dimensions are not supported. Please choose a smaller file.

TeamRaiser Address Book API Error Codes

The following error codes may be returned by calls to the client-only TeamRaiser Address Book API (CRAddressbookAPI).

Code	Description
901	The email provided matches an existing email address.
902	Some required fields were either missing or invalid.
903	A constituent ID is required.
904	You do not have permission to perform the requested operation.
905	A group with the same name already exists.

Logging API Calls

API logging is a useful tool for developing applications using Convio Open APIs or for diagnosing problems. The API logs record the Convio server's perspective of incoming API calls, the steps the system takes to process them, and the results returned or errors encountered. Logging can be disabled, or logs can be configured to capture no API calls, all calls, all calls with additional debugging information, or only those calls that return errors.

To configure API logging or to download and view log files, log in to the Convio website as a site administrator and follow the steps described in *Site Configuration*.

Client and Server APIs

Client versus Server APIs

Some Convio Open APIs provide separate Client and Server versions, and some API methods are available using Client or only Server access.

Convio Server APIs allow trusted third-party web servers to proxy requests from pages hosted on their site to the Convio web server:

- Web page from site A makes an AJAX call to a service on site A using an XMLHttpRequest. Since this is within the same origin, the browser's security policies allow it.

- The service on site A acts as a proxy and forwards the call to a Convio Server API, which then responds.
- The service on site A then replies to the browser with the response it got from the Convio Server API.

Server APIs generally act with administrator level privileges, whereas Client APIs generally act with the privileges of an individual logged-in user or as the "anybody" (i.e. not logged-in) user.

Server APIs require API Administrator credentials, passed as login_user and login_password on each request (see *API Administrator*). They also enforce IP address restrictions to verify that the request is coming from a trusted origin. Client APIs may operate in the context of an anonymous user or as a logged-in constituent, depending upon whether information they modify or return would be visible to the general public. Client APIs can operate in the context of a specific end user to read or modify personal information only if a valid sso_auth_token authentication token is passed in the request.

Convio Server APIs are fairly easy to implement, and can be implemented using any server-side programming language. Some frameworks (like Google pages or IGoogle) have server-side proxy features built into their JavaScript libraries.

On the other hand, using Convio Server APIs requires server-side programming, which may not be possible in some environments. They do not establish a connection between the user's browser and Convio, so it can be difficult to create a seamless experience if the user can also navigate to the Convio site directly. Data has to pass through both systems creating potential security exposures. This approach cannot be used for systems that must meet PCI Data Security Standards.

For this reason, Convio provides no support for Server APIs where users' sensitive information (e.g. passwords or credit card numbers) might pass from the browser to the third-party server and then on to the Convio server, such as for example the `Donation API`.

API Administrator Accounts for Server API Access

Convio server APIs require login_name and login_password on each API call. You must create an account in the API Administrator security group in order to use Convio server APIs, and pass the login credentials for this account in each API call.

To create or update Convio API Administrator accounts:

1. Log in to Convio as a site administrator.
2. Select **Site Options** from the **Setup** menu.
3. Click the **Open API Configuration** tab.
4. In the "Configure API to allow server access" section of the "Open API Configuration" screen, click the **Edit server API configuration** link.
5. Under **Manage API Administrative Accounts**, add at least one API Administrator account.

If you intend to allow multiple individuals or organizations to integrate with your site using these APIs, you should consider creating a Convio API Administrator account for each of them.

Members of the API Administrators group have only one permission: Use Convio APIs.

Note: Accounts in the API Administrators group can only call Convio server API methods; they cannot log in to the Convio Web site or perform any other administrative function.

IP White List Security

Server API calls must originate from trusted external systems. A Convio site administrator must grant calling systems access by adding them to the Convio API IP White List using the Open API Configuration application.

Detailed instructions on configuring Server API access through the White List appear in *Site Configuration*, and also the in Convio Site Administrator Help pages.

Server Authentication using Convio APIs

Authenticating Convio Server API calls is a two-stage process.

1. The IP address of the server invoking the API call will be checked against the list of allowed IP ranges, per the Convio Site Configuration parameters.
2. If the API is being invoked from a server in the allowed IP range, then the login_name and login_password parameters will be used to authenticate with the Convio site.
3. Each API or method may verify that the API Administrator account specified by the login_name and login_password parameters has the correct permission to access that particular API before authorizing access. See the documentation for the individual API or method for information on the necessary permissions.

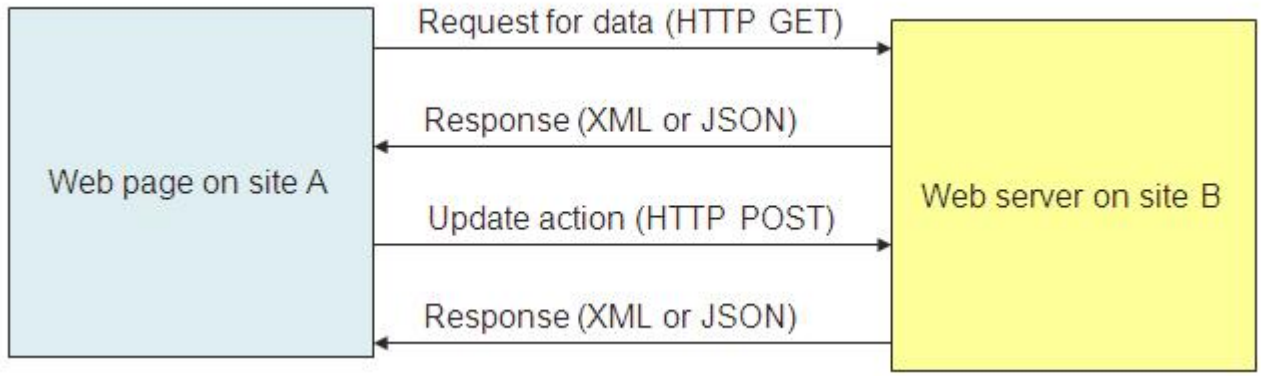
If the above succeeds, the caller will be authenticated and the API call will proceed. If any of the above fail, then the call will not be authenticated and an error message will be returned.

Cross-Domain Access

Cross-domain Access

At its simplest, cross-domain access involves getting data from one web site and displaying it within a page from a different web site.

In more complex scenarios, it involves constructing business process that both read and update data on different web sites including sensitive data that may be tied to the specific user who is performing the transaction.



Example: A page served from site A displays information about a cause.

- Site A asks for the user's name and address and then retrieves the name of the user's legislators from site B and displays that information.
- If the user chooses to send a letter, Site A passes the information to site B, where the letter is actually sent.
- Site B responds with an acknowledgment message or error messages if the letter could not be sent, and site A displays that information.

Cross-domain access can be difficult to manage because it must be handled securely. When a request comes to site B from a page on site A, it looks to site B as if it came directly from the browser of the user accessing site A. The content from a web server is a mixture of instructional code (e.g. JavaScript) and data (e.g. HTML content). If the browser gave different sites access to each other's content, a malicious site could steal or manipulate data purportedly coming from a trusted site. To prevent this, web browsers enforce a "same origin" policy:

- The "same origin" policy is a limitation in all browsers that prevents a document or script loaded from one "origin" from getting or setting properties of a document from a different "origin".
- "Origin" means the combination of domain, protocol (HTTP or HTTPS) and port, so it applies to content that is from the same domain but some is served securely and some is not.
- This policy prevents hostile code from one site from "taking over" or manipulating documents from another. Without it, JavaScript from a hostile site could do any number of undesirable things such as snoop key presses while a user logs in to a site in a different window, hijack a session with the user's online banking site and insert spurious transactions, steal login cookies from other domains, and so on.

Convio API methods support the *W3C Cross-Origin Resource Sharing (CORS)* specification for client-side cross-origin requests. You must white-list any allowed origins for cross-origin requests in your Convio site configuration.

JavaScript/AJAX Programming

Calling Convio APIs in JavaScript

You can use native JavaScript XMLHttpRequest (XHR) or the AJAX methods provided by JavaScript frameworks such as jQuery, YUI, or Dojo to call Convio Open API methods.

Convio APIs support the W3C Cross-Origin Resource Sharing (CORS) specification for client-side cross-origin requests. You must add the domain of each authorized website to the JavaScript/Flash Allowed Domains white list on your Convio site in order to enable cross-origin requests from pages hosted on those sites:

1. From the **Setup** menu, select **Site Options**.
2. Click the **Open API Configuration** tab.
3. Next to "Configure Javascript/Flash access to Open APIs" click the **Edit Javascript/Flash configuration** link.
4. Under **1. Allow JavaScript/Flash API from these domains** enter the list of domains you will allow to host cross-origin requests.

Note: Note: With support for CORS, the Convio Cross-domain JavaScript API Library located in the the `/api` directory on your Convio website has been deprecated. You should not use this library for new development.

JavaScript Examples

The following is a basic API call made using the native JavaScript XMLHttpRequest object:

```
var  uri = "https://secure2.convio.net/MY-ORG/site/CRConsAPI",
 postdata = "method=listUserFields&v=1.0&api_key=MY-API-KEY";

function makeXhrRequest () {
   xhr = new XMLHttpRequest();
   xhr.open('POST', uri, true);
   xhr.onreadystatechange=function() {
     if (xhr.readyState == 4) {
        console.log(xhr.status);
        console.log(xhr.responseText);
     }
   };
   xhr.setRequestHeader('Content-type','application/x-www-form-urlencoded');
   xhr.send(postdata);
}
```

Using the jQuery framework, the same API call looks like this:

```
var  uri = "https://secure2.convio.net/MY-ORG/site/CRConsAPI",
 postdata = "method=listUserFields&v=1.0&api_key=MY-API-KEY";

function makeJqueryRequest() {
  $.post(uri, postdata, function(data, textStatus, jqxhr){
     console.log(data);
     console.log(textStatus);
     console.log(jqxhr);
  });
}
```

This is the same API call using the YUI framework:

```
var  uri = "https://secure2.convio.net/MY-ORG/site/CRConsAPI",
 postdata = "method=listUserFields&v=1.0&api_key=MY-API-KEY";
```

```
function makeYuiRequest() {
   YAHOO.util.Connect.asyncRequest(
      'POST',
      uri,
      {
         success : function(o) {
            console.log(o.responseText);
     },
         failure : function(o) {
            console.log(o.responseText);
     },
         args : {}
      },
      postdata );
}
```

Finally, here is the same API call using the Dojo framework:

```
var  uri = "https://secure2.convio.net/MY-ORG/site/CRConsAPI",
 postdata = "method=listUserFields&v=1.0&api_key=MY-API-KEY";

function makeDojoRequest() {
   dojo.xhrPost({
      url: uri,
      postData: postdata, //consider content: or form: instead of postData:
      load: function(response) {
         console.log(response);
      }
   });
}
```

These examples demonstrate basic communication with Convio Open API methods through various JavaScript frameworks. You should refer to the reference documentation of whichever framework you choose for additional examples and information on parsing response data, handling errors, and serializing form data using the framework.

Adobe Flash Cross-domain Policy

Adobe Flash Player uses its own security model. Flash Player default behavior is that cross-domain access is restricted by the same origin policy. Individual web sites can override this default behavior. To do so, the web site administrator creates a set of policy definitions in the site's crossdomain.xml file. The crossdomain.xml file lists the other domains that can access resources at or below the level where file exists. The domain being checked is the domain of the web page hosting the Flash SWF file or movie, not the domain that the Flash SWF came from. The crossdomain.xml file can also specify ports (e.g. port 80 for HTTP protocol, vs. port 443 for HTTPS) and can specify whether the Flash movie had to be retrieved securely.

Unless a domain is used exclusively for API access, you should not open up access with a root-level crossdomain.xml file, for security reasons. If the policy file is not at the root level of the web server, clients must explicitly load the relevant policy files.

Convio can be configured to automatically create a crossdomain.xml policy file for each of the APIs so that they can be accessed using Flash from other domains. To load the policy file for the CRConsAPI you would use a URL similar to this one:

```
https://secure2.convio.net/organization/site/CRConsAPI/crossdomain.xml
```

The list of sites allowed access by this policy is configured as a white list in the Convio Open API Configuration settings screen. To configure it, log in to Convio as a site administrator, and from the Settings menu, select Site Options. When the Site Options page loads, click the Open API Configuration tab.

The policy for each API must be loaded separately by each Flash SWF that references it, and calls to the API from within Flash must include a trailing slash after the servlet name (e.g. `CRConsAPI/?v=1.0`, not `CRConsAPI?v=1.0`).

Single Sign-On

Single Sign On (Convio as client)

Single-sign on where the user is authenticated by the caller and logged in to Convio as the authentication client uses *getSingleSignOnToken* and *singleSignOn*.

The `getSingleSignOnToken` method is a server-only API that must be called from a trusted host. The host's IP address range must be specified in the Server API IP White List on the Convio server, and the call must include login credentials for an API Administrator account.

The Caller passes in a cons_id or member_id for a constituent in the Convio database and gets back a token that can be used to login as that user. The token expires after 15 minutes.

The `singleSignOn` method is a client-only API called from the constituent's browser with the token returned by `getSingleSignOnToken`, typically by throwing a redirect or linking rather than via AJAX. Additional parameters include a redirect to send the browser to after the login is completed.

1. Partner site presents a login form that submits username and password to the partner site, authenticates the user, logs him in and retrieves the user's Convio ID or Member ID from its database.
2. Partner site calls the `getSingleSignOnToken` API, passing the ID that it retrieved.
3. Normally, the partner site would redirect to its own page (say http://www.example.org/index.html), but instead sends a redirect to the Convio `singleSignOn` API: `http://member.example.org/site/CRConsAPI?method=singleSignOn&sso_auth_token=`*`(value of 'token' returned by getSingleSignOnToken)`*`&redirect=http://www.example.org/index.html` The browser will make a request to the Convio system, which will log the user in, push a session cookie and throw a 302 redirect back to the partner site.

Subsequent requests to the Convio site will connect as the user because of the session cookie that was pushed earlier. Additional client-side API requests can include sso_auth_token to verify that the request is coming from a trusted system.

Single Sign On (Convio as master)

Single Sign On (Convio as master) APIs permit a browser-based web application hosted on an external server to use Convio as the master authentication server in a single-sign-on relationship. Typical criteria for using Convio as master are:

- Integrating with a third-party CMS that does not have rich personalization capabilities.
- Integrating with an online community or other web application that will cater to a subset of your online users (the Convio database will be a superset of your online users).

In general, when a new user comes to any of your web applications, you may want to log them in or register them with Convio for email marketing, and use Convio to manage the account.

To establish a single sign-on session using Convio as the master authenticator, use either *login* or *authenticateUser*. Before logging the user in, you may use *loginTest* to check the user's status. The `loginTest` method returns a cons_id if the user is already logged in or could be logged in because of a "remember me" cookie.

Note: Note: loginTest should be invoked on an insecure channel since that is the most likely channel for a session already established.

Use *logout* to end a user's logged-in session on the Convio server.

See also *Single Sign On (Convio as client)* for information on implementing single sign-on using the external system as master authenticator.

To maintain the logged-in Convio session alive beyond its 15-minute time-out, the partner web site only needs to include an image tag that references a servlet on the Convio powered site:

```
<img src=http://www.foo.org/site/PixelServer />
```

This will render a 1x1 clear GIF image that will maintain the session while the user remains connected.

Authentication Tokens

About Authentication Tokens

An authentication token is a piece of information that is generated by the Convio server and passed via a known secure channel that can then be echoed back as part of an HTTP request to verify the authenticity of the caller.

An authentication token is required for any client API that accesses personal data about a constituent that is not publicly visible. The authentication token is required because reliance on only a session cookie to establish the identity of the caller could potentially open the APIs to cross-site request forgeries. This extra level of security ensures that a constituent's personal data cannot be accessed or modified by code running on a malicious site.

There are two types of tokens that can be used for this form of authentication. The first is essentially a "nonce" or information-less token that is stored in the user's session. This can be rendered by the Convio engine in content pages using the `[[S86]]` session tag, or can be accessed from a trusted domain via the Convio JavaScript API. This token must be sent back to the same user session that created it in order to authenticate the request. The second is a single-sign-on or SSO token. This token is generated by the Convio system and passed back via a server-side API call that is secure by virtue of IP address white-listing. This token is associated with a particular Convio constituent and can be sent to an anonymous session in order to log in as that constituent.

Using the Authentication Token on a Page Hosted by Convio

When building a form on a Convio web site in PageBuilder, you can insert the "Convio API: Authentication Token" component from the Components drop-down menu. This component is equivalent to the S86 tag in the page HTML source. The component or tag will be expanded to a hidden form element that will generate and pass the correct token when the form is submitted.

If you use scripting to send the request instead of form submission, your script must get the token value from the hidden field and submit it with API calls using the parameter name auth. You can also render just the value of the token (as opposed to the entire hidden input element) by using the tag [[S86:true]]. In this example, "true" is the value of the "token only" flag, indicating that only the token value should be rendered. Although the token is visible when viewing the page source, it adds security because it is temporary and not valid for any other user.

Applying this authentication token from a Convio hosted and trusted page ensures the most secure and reliable experience for your constituents. This component must be included in the following Web Client calls:

- getUser
- getUserGroups
- getUserInterests
- update
- createOrUpdate

In addition, most TeamRaiser API methods require the authentication token.

Note: Instead of using the WYSIWYG Components menu, you can also enter the tag directly using one of the following:

Bracket Syntax:

```
[[S86]]
```

XML Syntax:

```
<convio:session name="86"/>
```

Using the Authentication Token when Creating a Page that is External to Convio

When creating a page in an outside application, you can use the SRConsAPI:getSingleSignOnToken server-side method to return a single sign-on "token" (SSO token) that can then be used in client-side requests to automatically log a user into the system.

This approach is appropriate in the case where a Web site external to Convio has authenticated the site visitor and wants to direct the user's browser to the Convio system and have them automatically logged in to the Convio system. The external web server, after verifying the identity of the visitor, calls SRConsAPI:getSingleSignOnToken to get a token that will authenticate the source of the request when sent by the visitor's browser in a subsequent HTTP request. This token is only valid for the specific user and only for a limited time (typically, a few minutes).

Instead of using this token as the auth parameter, you should add this token as the sso_auth_token to any API request.

Making Server-to-Server Calls

Server-to-server calls do not require authentication tokens because you must enter the IP addresses of the external sites that you will permit to make calls when you set up your site for API access.

Calling the API from an Insecure Page

When calling the Convio API from an insecure page (a page not served via SSL) in Convio, the URL must be wrapped in a T2 tag to add the JservSessionId parameter. For example:

```
<form action="[[T2:https://secure2.convio.net/convio/site/CRConsAPI]]">
```

This tag ensures that the user's authenticated session information is transferred from the insecure server to the secure server.

Note: It is simpler for an administrator to build the page as a secure page (there is a check box in PageBuilder to make the page an SSL page). However, this may not always be feasible.

API Reference

Address Book API

CRAddressBookAPI
`https://secure2.convio.net/organization/site/CRAddressBookAPI`

Description

The Address Book API is a Client API that allows logged-in TeamRaiser participants to manage their email address books. These APIs require an *Authentication Token*, representing a logged-in user session for the TeamRaiser participant, either from an embedded [[S86]] session tag in a Convio web page, or returned by the appropriate *Single Sign-On API*.

The Address Book API exposes the following methods:

addAddressBookContact

Adds a single contact to your address book.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	Yes.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=addAddressBookContact &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&email=`*value* [`&first_name=`*value*] [`&last_name=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

email

Required. The contact's email address.

Type xsd:string.

first_name

Optional. The contact's first name.

Type xsd:string.

last_name

Optional. The contact's last name.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<addAddressBookContactResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <addressBookContact>
    <id>101</id>
    <firstName>User1First</firstName>
    <lastName>User1Last</lastName>
    <email>user1@example.com</email>
    <group>
      <id>201</id>
      <name>Group 1</name>
    </group>
  </addressBookContact>
</addAddressBookContactResponse>
```

JSON response

```
{"addAddressBookContactResponse":{"addressBookContact":
{"id":"101","lastName":"User1Last","email":"user1@example.com","group":
{"id":"201","name":"Group 1"},"firstName":"User1First"}}}
```

addAddressBookContacts

Adds a list of name and email addresses to your address book.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=addAddressBookContacts &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&contacts_to_add=`*value* [`&auth=`*value*][`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

contacts_to_add

Required. Comma-separated list of names and addresses. e.g. "Bob Smith" <bob@example.com>, "Kelly Smith" <kelly@example.com>

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getAddressBookContactsResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <addressBookContact>
    <id>100</id>
    <firstName>User0First</firstName>
    <lastName>User0Last</lastName>
    <email>user0@example.com</email>
    <group>
      <id>200</id>
      <name>Group 0</name>
    </group>
    <group>
      <id>200</id>
      <name>Group 0</name>
    </group>
  </addressBookContact>
  <addressBookContact>
    <id>101</id>
    <firstName>User1First</firstName>
    <lastName>User1Last</lastName>
    <email>user1@example.com</email>
    <group>
      <id>201</id>
      <name>Group 1</name>
    </group>
  </addressBookContact>
</getAddressBookContactsResponse>
```

JSON response

```
{"getAddressBookContactsResponse":{"addressBookContact":
[{"id":"100","lastName":"User0Last","email":"user0@example.com","group":
[{"id":"200","name":"Group 0"},{"id":"200","name":"Group
 0"}],"firstName":"User0First"},
{"id":"101","lastName":"User1Last","email":"user1@example.com","group":
{"id":"201","name":"Group 1"},"firstName":"User1First"}]}}
```

addAddressBookGroup

Creates a new group for your address book contacts.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=addAddressBookGroup &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&group_name=`*value* [`&contact_ids=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

group_name

Required. The name of the group.

Type xsd:string.

contact_ids

Optional. Optional comma-separated list of contact IDs to add to the new group.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<addAddressBookGroupResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <addressBookGroup>
    <id>123</id>
    <name>Sample Group Name</name>
  </addressBookGroup>
</addAddressBookGroupResponse>
```

JSON response

```
{"addAddressBookGroupResponse":{"addressBookGroup":{"id":"123","name":"Sample
 Group Name"}}}
```

addContactsToGroup

Adds contacts to an existing group.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=addContactsToGroup &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&group_id=`*value* [`&contact_ids=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

group_id

Required. A unique ID of the group to add contacts to.

Type xsd:nonNegativeInteger.

contact_ids

Optional. Comma-separated list of contact IDs to add to the specified group.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<addContactsToGroupResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <addressBookGroup>
    <id>123</id>
    <name>Sample Group Name</name>
  </addressBookGroup>
  <contactId>100</contactId>
  <contactId>101</contactId>
  <contactId>102</contactId>
```

```
  <contactId>103</contactId>
  <contactId>104</contactId>
</addContactsToGroupResponse>
```

JSON response

```
{"addContactsToGroupResponse":{"contactId":
["100","101","102","103","104"],"addressBookGroup":{"id":"123","name":"Sample
 Group Name"}}}
```

addressBookImportOAuthCallback

Handles the callback from an Open Authentication provider and fetches the user's address book

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=addressBookImportOAuthCallback &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&import_job_id=`*value* `&oauth_token=`*value* `&oauth_verifier=`*value*

Usage Notes

Prior to calling this method, you must successfully call *startOnlineAddressBookImport* and redirect the user to the authentication URL returned by the host provider.

This method passes the returned OAuth credentials to the hosting provider along with a request to begin the download job. After the user authenticates with the hosting provider and gives consent for your app to access his address book, the hosting provider redirects to the URL in the callback_url parameter of *startOnlineAddressBookImport*. The callback_url may specify this method directly, or may specify a page on your web server containing script that calls this method (for an example, see the callback handler page in Convio TeamRaiser Participant Center, `/pc2/address_book_import_oauth_callback_handler.html`).

Note: By default, your site is configured to use Convio's OAuth credentials when requesting addresses from Google or Yahoo. As a result, users will be notified that "Convio" is requesting permission to access their address book. If you want your organization's name to appear instead, you must obtain your own OAuth credentials and configure your site to use them.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

import_job_id

Required. ID of an address book import job.

Type xsd:integer.

oauth_token

Required. Open Authentication token.

Type xsd:token.

oauth_verifier

Required. Open Authentication verifier.

Type xsd:token.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<addressBookImportOAuthCallbackResponse xsi:schemaLocation="http://
convio.com/crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd"
 xmlns="http://convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">OK</addressBookImportOAuthCallbackResponse>
```

JSON response

```
{"addressBookImportOAuthCallbackResponse":{"content":"OK"}}
```

deleteAddressBookContacts

Deletes one or more contacts from the owner's address book. Cannot delete contacts shared by Teamraiser team members.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=deleteAddressBookContacts &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&contact_ids=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

A constituent's Address Book is shared across several applications in Convio, including TeamRaiser, Personal Fundraising, and Personal Events.

A TeamRaiser Participant who is also a Personal Fundraising Champion will see the same contacts in their Address Book when they are sending mail through their Participant Center or their Champion Center.

This method can be used to remove contacts from the constituent's address book. However, it cannot delete contacts that are shared between Teamraiser Team members (such as a donor that gave a gift on behalf of the team) - in order to delete those contacts, you must use the TeamRaiser API's *deleteTeamraiserAddressBookContacts* method.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

contact_ids

Optional. Comma-separated list of contact IDs to delete.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<deleteAddressBookContactsResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <contactId>101</contactId>
  <contactId>102</contactId>
  <contactId>103</contactId>
  <errors>false</errors>
</deleteAddressBookContactsResponse>
```

JSON response

```
{"deleteAddressBookContactsResponse":{"errors":"false","contactId":
["101","102","103"]}}
```

deleteAddressBookGroups

Deletes one or more Address Book groups.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=deleteAddressBookGroups &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&group_ids=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

Given one or more group_id values, this method will delete the Address Book groups specified by the IDs. Deleting a group does not delete the group's member contacts. To delete contacts, use *deleteAddressBookContacts*.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

group_ids

Required. Comma-separated list of group IDs.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<removedAddressBookGroupResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <group_id>123</group_id>
  <group_id>234</group_id>
</removedAddressBookGroupResponse>
```

JSON response

```
{"removedAddressBookGroupResponse":{"group_id":["123","234"]}}
```

getAddressBookContacts

Provides a list of Address Book contacts for the logged in user.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=getAddressBookContacts &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*][`&list_ascending=`*value*][`&list_category_id=`*value*][`&list_filter_column=`*value*][`&list_filter_text=`*value*][`&list_page_offset=`*value*][`&list_page_size=`*value*][`&list_record_offset=`*value*][`&list_sort_column=`*value*][`&auth=`*value*][`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getAddressBookContactsResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <addressBookContact>
```

```
    <id>101</id>
    <firstName>User1First</firstName>
    <lastName>User1Last</lastName>
    <email>user1@example.com</email>
    <group>
      <id>201</id>
      <name>Group 1</name>
    </group>
  </addressBookContact>
  <addressBookContact>
    <id>102</id>
    <firstName>User2First</firstName>
    <lastName>User2Last</lastName>
    <email>user2@example.com</email>
    <group>
      <id>202</id>
      <name>Group 2</name>
    </group>
    <group>
      <id>202</id>
      <name>Group 2</name>
    </group>
  </addressBookContact>
</getAddressBookContactsResponse>
```

JSON response

```
{"getAddressBookContactsResponse":{"addressBookContact":
[{"id":"101","lastName":"User1Last","email":"user1@example.com","group":
{"id":"201","name":"Group 1"},"firstName":"User1First"},
{"id":"102","lastName":"User2Last","email":"user2@example.com","group":
[{"id":"202","name":"Group 2"},{"id":"202","name":"Group
 2"}],"firstName":"User2First"}]}}
```

getAddressBookGroups

Provides a list of Address Book group names and group IDs for the logged in user.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=getAddressBookGroups &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*][`&count_contacts=`*value*][`&auth=`*value*][`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

count_contacts

Optional. Whether or not to include a count of contacts belonging to a group.

Type xsd:boolean.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getAddressBookGroupsResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <group>
    <id>1000</id>
    <name>Sample Group 1</name>
    <contactsCount>3</contactsCount>
  </group>
  <group>
    <id>1001</id>
    <name>Sample Group 2</name>
    <contactsCount>4</contactsCount>
  </group>
  <group>
    <id>1002</id>
    <name>Sample Group 3</name>
    <contactsCount>5</contactsCount>
  </group>
</getAddressBookGroupsResponse>
```

JSON response

```
{"getAddressBookGroupsResponse":{"group":[{"id":"1000","name":"Sample
 Group 1","contactsCount":"3"},{"id":"1001","name":"Sample
 Group 2","contactsCount":"4"},{"id":"1002","name":"Sample Group
 3","contactsCount":"5"}]}}
```

getAddressBookImportContacts

Gets the contacts from an address book import job

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=getAddressBookImportContacts &api_key=`*value* `&v=`*value*

[`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&import_job_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method returns a list of contact data imported from an online address book such as Gmail or Yahoo. Prerequisite to calling this method, you must obtain a job ID from *startOnlineAddressBookImport* and *addressBookImportOAuthCallback*, and the *getAddressBookImportJobStatus* method must reflect that the import job has completed successfully.

After retrieving contacts from the hosting provider with this method, call *importAddressBookContacts* to add the returned contacts to the constituent's address book. For usage details refer to the Usage Notes for *startOnlineAddressBookImport*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

import_job_id

Required. ID of an address book import job.

Type xsd:integer.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getAddressBookImportContactsResponse xsi:schemaLocation="http://convio.com/
crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <contact>
    <firstName>Adam</firstName>
    <lastName>Adams</lastName>
    <email>aadams@address.net</email>
  </contact>
  <contact>
    <firstName>Brian</firstName>
    <lastName>Brians</lastName>
    <email>bbrians@address.net</email>
  </contact>
  <contact>
    <firstName>Charlie</firstName>
    <lastName>Charles</lastName>
    <email>ccharles@address.net</email>
  </contact>
</getAddressBookImportContactsResponse>
```

JSON response

```
{"getAddressBookImportContactsResponse":{"contact":
[{"lastName":"Adams","email":"aadams@address.net","firstName":"Adam"},
{"lastName":"Brians","email":"bbrians@address.net","firstName":"Brian"},
{"lastName":"Charles","email":"ccharles@address.net","firstName":"Charlie"}]}}
```

getAddressBookImportJobStatus

Gets the status of an address book import job

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=getAddressBookImportJobStatus &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&import_job_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method returns the status of an active import job. Prerequisite to calling this method, you must obtain a job ID from *startOnlineAddressBookImport*.

This method will return "PENDING" status until the constituent authenticates with the hosting provider and grants access to his address book, resulting in a successful call to *addressBookImportOAuthCallback*.

Once the download completes, use *getAddressBookImportContacts* to return the downloaded data.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

import_job_id

Required. ID of an address book import job.

Type xsd:integer.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getAddressBookImportJobStatusResponse xsi:schemaLocation="http://convio.com/
crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <jobStatus>PENDING</jobStatus>
  <events>
    <event>Received authorization</event>
    <event>Importing Contacts</event>
  </events>
</getAddressBookImportJobStatusResponse>
```

JSON response

```
{"getAddressBookImportJobStatusResponse":{"jobStatus":"PENDING","events":
{"event":["Received authorization","Importing Contacts"]}}}
```

importAddressBookContacts

For a given list of contacts in CSV format ("first name", "last name", "email address") this REST method will: 1) Add and return new contacts that don't match email, first name or last name 2) Return a list of potential duplicate contacts that match either email OR first and last name 3) Return a list of full duplicate contacts that match email, first name and last name 4) Return a list of contacts that could not be added due to errors

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=importAddressBookContacts &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&contacts_to_add=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

Like *addAddressBookContacts*, this method allows you to add contacts to the logged-in constituent's address book, but returns a different response structure and provides a higher degree of duplicate and error checking.

For a supplied list of addresses, this method returns: a list of new contacts, a list of potential duplicates, a list of duplicates, and a list of any contacts that could not be added due to errors (e.g. email address validation failure).

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

contacts_to_add

Required. Comma-separated list of names and addresses. e.g. "Bob Smith" <bob@example.com>, "Kelly Smith" <kelly@example.com>

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<importAddressBookContactsResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <savedContact>
    <contact>
      <id>1001001</id>
      <firstName>Eric</firstName>
      <lastName>Ericson</lastName>
      <email>eericson@address.net</email>
    </contact>
  </savedContact>
  <savedContact>
    <contact>
      <id>1001002</id>
      <firstName>Frank</firstName>
      <lastName>Franklin</lastName>
      <email>ffranklin@address.net</email>
    </contact>
  </savedContact>
  <duplicateContact>
    <contact>
      <firstName>Adam</firstName>
      <lastName>Adams</lastName>
      <email>aadams@address.net</email>
    </contact>
    <duplicateContact>
      <id>1001</id>
      <firstName>Adam</firstName>
      <lastName>Adams</lastName>
      <email>aadams@address.net</email>
    </duplicateContact>
  </duplicateContact>
  <duplicateContact>
    <contact>
      <firstName>Brian</firstName>
      <lastName>Brians</lastName>
      <email>bbrians@address.net</email>
    </contact>
    <duplicateContact>
      <id>1002</id>
      <firstName>Brian</firstName>
      <lastName>Brians</lastName>
      <email>bbrians@address.net</email>
    </duplicateContact>
  </duplicateContact>
  <potentialDuplicateContact>
    <contact>
      <firstName>Charlie</firstName>
      <lastName>Charles</lastName>
      <email>ccharles@address.net</email>
    </contact>
```

```
    <duplicateContact>
      <id>1003</id>
      <firstName>Charlie</firstName>
      <lastName>Charles</lastName>
      <email>ccharlessomeotheraddress.net</email>
    </duplicateContact>
  </potentialDuplicateContact>
  <potentialDuplicateContact>
    <contact>
      <firstName>David</firstName>
      <lastName>Davidson</lastName>
      <email>ddavidson@address.net</email>
    </contact>
    <duplicateContact>
      <id>1004</id>
      <firstName>Other</firstName>
      <lastName>Name</lastName>
      <email>ddavidson@address.net</email>
    </duplicateContact>
  </potentialDuplicateContact>
  <errorContact>
    <contact>
      <firstName>Greg</firstName>
      <lastName>Gregory</lastName>
      <email>ggreg@address</email>
    </contact>
    <error>ggreg@address is not a valid email address</error>
  </errorContact>
</importAddressBookContactsResponse>
```

JSON response

```
{"importAddressBookContactsResponse":{"potentialDuplicateContact":
[{"duplicateContact":
{"id":"1003","lastName":"Charles","email":"ccharlessomeotheraddress.net","firstName":"Cha
{"lastName":"Charles","email":"ccharles@address.net","firstName":"Charlie"}},
{"duplicateContact":
{"id":"1004","lastName":"Name","email":"ddavidson@address.net","firstName":"Other"},"cont
{"lastName":"Davidson","email":"ddavidson@address.net","firstName":"David"}}],"savedConta
[{"contact":
{"id":"1001001","lastName":"Ericson","email":"eericson@address.net","firstName":"Eric"}},
{"contact":
{"id":"1001002","lastName":"Franklin","email":"ffranklin@address.net","firstName":"Frank"
{"error":"ggreg@address is not a valid email address","contact":
{"lastName":"Gregory","email":"ggreg@address","firstName":"Greg"}},"duplicateContact":
[{"duplicateContact":
{"id":"1001","lastName":"Adams","email":"aadams@address.net","firstName":"Adam"},"contact
{"lastName":"Adams","email":"aadams@address.net","firstName":"Adam"}},
{"duplicateContact":
{"id":"1002","lastName":"Brians","email":"bbrians@address.net","firstName":"Brian"},"cont
{"lastName":"Brians","email":"bbrians@address.net","firstName":"Brian"}}]}}
```

parseCsvContacts

Parses a CSV file of contact data and suggests a probable column mapping for import into the Convio address book.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=parseCsvContacts &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&contacts_upload_file=`*value* [`&file_encoding=`*value*]

Usage Notes

Given a contacts_upload_file in comma-separated value (CSV) format and an optional file_encoding, this method attempts to parse and return contact information from the file.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

contacts_upload_file

Required. CSV contacts file

file_encoding

Optional. CSV contacts file encoding

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<parseCsvContactsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <proposedCharacterEncoding>ISO-8859-15</proposedCharacterEncoding>
  <columnHeaderLabels>
    <columnHeaderLabel>first_name</columnHeaderLabel>
    <columnHeaderLabel>middle_name</columnHeaderLabel>
    <columnHeaderLabel>last_name</columnHeaderLabel>
    <columnHeaderLabel>email</columnHeaderLabel>
    <columnHeaderLabel>phone_number</columnHeaderLabel>
  </columnHeaderLabels>
  <proposedMapping>
    <firstNameColumnIndex>0</firstNameColumnIndex>
    <lastNameColumnIndex>2</lastNameColumnIndex>
    <emailColumnIndex>3</emailColumnIndex>
  </proposedMapping>
  <confidenceLevel>HIGH</confidenceLevel>
  <csvDataRows>
    <csvDataRow>
      <csvValue>John</csvValue>
      <csvValue>Scott</csvValue>
      <csvValue>Smith</csvValue>
      <csvValue>jscott@fakemail.com</csvValue>
      <csvValue>(512) 492-1123</csvValue>
    </csvDataRow>
    <csvDataRow>
      <csvValue>Betsy</csvValue>
      <csvValue xsi:nil="true"/>
      <csvValue xsi:nil="true"/>
      <csvValue>betsy@fakemail.com</csvValue>
```

```
      <csvValue>(214) 923-1928</csvValue>
    </csvDataRow>
    <csvDataRow>
      <csvValue>Juan</csvValue>
      <csvValue xsi:nil="true"/>
      <csvValue>DÃ¡vila, Jr.</csvValue>
      <csvValue>juan.davila@fakemail.com</csvValue>
      <csvValue xsi:nil="true"/>
    </csvDataRow>
  </csvDataRows>
</parseCsvContactsResponse>
```

JSON response


```
{"parseCsvContactsResponse":{"csvDataRows":{"csvDataRow":[{"csvValue":
["John","Scott","Smith","jscott@fakemail.com","(512) 492-1123"]},
{"csvValue":["Betsy",null,null,"betsy@fakemail.com","(214)
 923-1928"]},{"csvValue":["Juan",null,"DÃ¡vila,
 Jr.","juan.davila@fakemail.com",null]}]},"proposedMapping":
{"lastNameColumnIndex":"2","emailColumnIndex":"3","firstNameColumnIndex":"0"},"columnHeac
{"columnHeaderLabel":
["first_name","middle_name","last_name","email","phone_number"]},"confidenceLevel":"HIGH"
```


removeContactFromGroup

Removes a contact from an address book group.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=removeContactFromGroup &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&contact_id=`*value* `&group_id=`*value* [`&auth=`*value*][`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

contact_id

Required. The system-generated unique contact ID for the contact.

Type xsd:nonNegativeInteger.

group_id

Required. The ID of the group you wish to remove the contact from.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<removeContactFromGroupResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <group_id>123</group_id>
  <contact_id>45678</contact_id>
</removeContactFromGroupResponse>
```

JSON response

```
{"removeContactFromGroupResponse":{"contact_id":"45678","group_id":"123"}}
```

startOnlineAddressBookImport

Starts the process of importing an address book from an online email provider

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=startOnlineAddressBookImport &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&import_source=`*GMAIL* | *YAHOO* [`&callback_url=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

The methods: *startOnlineAddressBookImport*, *addressBookImportOAuthCallback*, *getAddressBookImportJobStatus*, and *getAddressBookImportContacts* methods use *OAuth* authentication to request a constituent's consent through a hosting provider such as Google Gmail or Yahoo to access the constituent's address book, and to download the contacts to Convio.

The *importAddressBookContacts* method may be used to import a comma-separated list of addresses into the constituent's Convio Addres Book, whether downloaded from an online provider using the methods above or exported from an email applications such as Microsoft Outlook or Mozilla Thunderbird. The *parseCsvContacts* method allows you to easily upload, analyze and parse CSV files from such applications.

This method requires an import_source (either GOOGLE or YAHOO), and a callback_url pointing to a page in your app that will accept the job Id, OAuth token and verifier from the hosting provider when the constituent grants consent. This method returns a `<jobId>` for the import job and the `<oauthUrl>` to the page on the host provider

that will verify the identity of the constituent and request permission to give your app access to their address book. Your app must redirect to or render the `<oauthUrl>` contents.

When the constituent grants access to their address book, the hosting provider will call the page you provided in the callback_url parameter. This page parses the import_job_id, oauth_token, and oauth_verifier parameters and calls *addressBookImportOAuthCallback* to begin downloading addresses from the hosting provider.

You can monitor the status of the download by calling *getAddressBookImportJobStatus* and specifying the job ID returned by this method. Once the download completes, your app must retrieve the downloaded addresses by calling *getAddressBookImportContacts* with the job ID, and then add the downloaded contacts to the constituent's Convio address book using *importAddressBookContacts*.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

import_source

Required. The online email service providing the address book.

Type xsd:string.

Options are:

- GMAIL
- YAHOO

callback_url

Optional. The url that will handle the oauth callback.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<startOnlineAddressBookImportResponse xsi:schemaLocation="http://convio.com/
crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <oauthUrl>http://www.google.com/oauth/49834908</oauthUrl>
  <jobId>1001</jobId>
</startOnlineAddressBookImportResponse>
```

JSON response

```
{"startOnlineAddressBookImportResponse":{"jobId":"1001","oauthUrl":"http://
www.google.com/oauth/49834908"}}
```

updateAddressBookContact

Updates an existing contact with new information.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=updateAddressBookContact &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&contact_id=`*value* `&email=`*value* `&first_name=`*value* `&last_name=`*value* [`&city=`*value*][`&phone=`*value*][`&state=`*value*][`&street1=`*value*][`&street2=`*value*][`&zip=`*value*][`&auth=`*value*][`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

contact_id

Required. The system-generated unique contact ID for the contact.

Type xsd:nonNegativeInteger.

email

Required. The contact's email address.

Type xsd:string.

first_name

Required. The contact's first name.

Type xsd:string.

last_name

Required. The contact's last name.

Type xsd:string.

city

Optional. The contact's city.

Type xsd:string.

phone

Optional. The contact's phone number.

Type xsd:string.

state

Optional. The contact's state or province.

Type xsd:string.

street1

Optional. The first line of the contact's street address.

Type xsd:string.

street2

Optional. The second line of the contact's street address, if needed.

Type xsd:string.

zip

Optional. The contact's postal or zip code.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateAddressBookContactResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
</updateAddressBookContactResponse>
```

JSON response

```
{"updateAddressBookContactResponse":{"success":"true"}}
```

updateAddressBookGroup

This method is used to update an existing Address Book group.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAddressBookAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAddressBookAPI?method=updateAddressBookGroup &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&group_id=`*value* `&group_name=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

Use this method to change the name of the address book group specified by group_id.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

group_id

Required. The ID of a specified group.

Type xsd:integer.

group_name

Required. The name of the group.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateAddressBookGroupResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <addressBookGroup>
    <id>123</id>
    <name>Updated Group Name</name>
  </addressBookGroup>
</updateAddressBookGroupResponse>
```

JSON response

```
{"updateAddressBookGroupResponse":{"addressBookGroup":
{"id":"123","name":"Updated Group Name"}}}
```

Advocacy API

CRAdvocacyAPI
`https://secure2.convio.net/organization/site/CRAdvocacyAPI`
SRAdvocacyAPI
`https://secure2.convio.net/organization/site/SRAdvocacyAPI`

Description

The Advocacy API permits you to query advocacy alerts, to query actions taken by a given constituent, or to allow a constituent to take action on an alert via the API.

The Convio Advocacy product must be installed and configured for your Convio client site in order to use the Advocacy APIs.

API Administrator accounts used for Server API calls to the Advocacy API must belong to an Administrative Security Group that has "Use Advocacy API" permission in the "Advocacy Management" permission type. The built-in "API Administrators" group has this permission by default.

Advocacy APIs expose the following methods:

getAdvocacyAlert

This method returns the information required for a given constituent to participate in an alert.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAdvocacyAPI	SRAdvocacyAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAdvocacyAPI?method=getAdvocacyAlert &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sso_auth_token=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&alert_id=`*value* `&alert_type=`*action* | *call* | *lte* [`&city=`*value*] [`&country=`*value*] [`&state=`*value*] [`&street1=`*value*] [`&street2=`*value*] [`&zip=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRAdvocacyAPI?method=getAdvocacyAlert & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& cons_id =`*value*] [`& error_redirect =`*value*] [`& redirect =`*value*] [`& response_format =`*xml* |*json*] [`& sign_redirects =`*value*] [`& source =`*value*] [`& sub_source =`*value*] [`& success_redirect =`*value*] [`& suppress_response_codes =`*value*] `& alert_id =`*value* `& alert_type =`*action* |*call* |*lte* [`& city =`*value*] [`& country =`*value*] [`& state =`*value*] [`& street1 =`*value*] [`& street2 =`*value*] [`& zip =`*value*]

Usage Notes

Call `getAdvocacyAlert` to obtain the information required by the *takeAction* API to participate in an alert.

You must specify both the alert_id and the alert_type of the alert you wish to retrieve, as alert_id is not guaranteed unique across types. This method returns suggested text, appropriate targets based on the calling constituent's federal and/or state legislative districts, and the available communication channels for each target.

This method returns values for title, first_name, last_name, street1, street2, city, state, zip, country, phone, and email parameters from the constituent profile if the calling constituent can be identified from a logged-in session or from the cons_id parameter passed in the server API call. It may also return other data from the constituent profile if the underlying alert includes survey questions for updating other fields.

This method also accepts address information as input, in which case the values you specify when calling the method override the values stored in the constituent profile (if they exist) for the purpose of determining the legislative district.

The following table describes how name and address information is used by this API method to determine targets based on state and federal legislative districting data:

Caller Context	Result
Client API in a logged-in user session, or Server API with valid cons_id.	If you do not pass address information, system will return the address information from the

Caller Context	Result
	constituent's record, if any, and will return only targets appropriate to that address. If you pass address information, the system will return targets applicable to the passed-in address instead.
Client API in an anonymous user session, or Server API without a cons_id.	If you do not pass address information, the system will only return response targets that are applicable to all constituents (e.g. President or Vice President), if any exist for the alert. If you do pass address information, the system will return any targets appropriate to the passed-in address.

Note: This method and the `takeAction` do not currently support the "vote" alert_type.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

alert_id

Required. Specify the alert.

Type xsd:positiveInteger.

alert_type

Required. Specify the alert type.

Type advocacyAlertType.

Options are:

- action
- call
- lte

city

Optional. Constituent city.

Type xsd:string.

country

Optional. Constituent country.

Type xsd:string.

state

Optional. Constituent state.

Type xsd:string.

street1

Optional. Constituent street 1.

Type xsd:string.

street2

Optional. Constituent street 2.

Type xsd:string.

zip

Optional. Constituent zip.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Optional. Specifies the constituent associated with the alert or action.

Type xsd:nonNegativeInteger.

sign_redirects

Optional. Specifies that redirect URLs should be signed with a timestamp and a hash of the query string arguments and a secret key.

Type xsd:boolean.

Default is false.

alert_id

Required. Specify the alert.

Type xsd:positiveInteger.

alert_type

Required. Specify the alert type.

Type advocacyAlertType.

Options are:

- action
- call
- lte

city

Optional. Constituent city.

Type xsd:string.

country

Optional. Constituent country.

Type xsd:string.

state

Optional. Constituent state.

Type xsd:string.

street1

Optional. Constituent street 1.

Type xsd:string.

street2

Optional. Constituent street 2.

Type xsd:string.

zip

Optional. Constituent zip.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getAdvocacyAlertResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <alert>
    <alertId>103</alertId>
    <type>action</type>
    <status>active</status>
    <priority>high</priority>
    <url>https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&amp;id=103</url>
    <interactionCount>1000002</interactionCount>
    <title>Alert Title</title>
    <thumbnail>https://secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/103</thumbnail>
    <internalName>Internal Name</internalName>
    <description>Description</description>
    <category categoryId="1">General</category>
    <issues>
      <issue issueId="1001">Issue1</issue>
      <issue issueId="1002">Issue2</issue>
    </issues>
    <restrictByState>
      <state>TN</state>
      <state>TX</state>
    </restrictByState>
    <modified>2013-02-03T05:18:23.651-06:00</modified>
    <publish>2013-02-03T05:18:23.651-06:00</publish>
    <expire>2013-02-03T05:18:23.650-06:00</expire>
    <targets>
      <target>Governor</target>
      <target>US Senator</target>
      <target>US Representative</target>
      <target>State Senator</target>
      <target>State Representative</target>
    </targets>
    <messageSubject>Subject</messageSubject>
    <messageSubjectEditable>optional</messageSubjectEditable>
    <messageGreeting>Dear</messageGreeting>
    <messageOpening>Message Opening Text</messageOpening>
    <messageBody>Message Body Text</messageBody>
    <messageBodyEditable>optional</messageBodyEditable>
    <messageClosing>Sincerely,</messageClosing>
    <messageSignature>Signature</messageSignature>
    <wordLimit>0</wordLimit>
    <position>none</position>
    <yeaCount>0</yeaCount>
    <nayCount>0</nayCount>
    <abstainCount>0</abstainCount>
  </alert>
  <questions>
    <question>
      <questionId>email</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value>user@example.com</value>
```

```
      <required>true</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>title</questionId>
      <questionType>MultiSingle</questionType>
      <text/>
      <value>Ms.</value>
      <required>true</required>
      <questionChoices>
        <choice>Mr.</choice>
        <choice>Ms.</choice>
        <choice>Dr.</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>first_name</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value>Jane</value>
      <required>true</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>last_name</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value>Public</value>
      <required>true</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>street1</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value>1234 Main St</value>
      <required>true</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>street2</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value/>
```

```
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>city</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value>AnyTown</value>
      <required>true</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>state</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value>DC</value>
      <required>true</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>zip</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value>20002</value>
      <required>true</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>country</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value/>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>phone</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value>555-1212</value>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
```

```
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>subject</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value>Abstain from H.R. 12345</value>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>body</questionId>
      <questionType>TextValue</questionType>
      <text/>
      <value>Please do not vote on this bill!</value>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>opt_in</questionId>
      <questionType>TrueFalse</questionType>
      <text/>
      <value>true</value>
      <required>false</required>
      <questionChoices>
        <choice>True</choice>
        <choice>False</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>anonymous</questionId>
      <questionType>TrueFalse</questionType>
      <text/>
      <value>false</value>
      <required>false</required>
      <questionChoices>
        <choice>True</choice>
        <choice>False</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>cc</questionId>
      <questionType>TrueFalse</questionType>
      <text/>
      <value>true</value>
      <required>false</required>
      <questionChoices>
```

```
        <choice>True</choice>
        <choice>False</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_1</questionId>
      <questionType>Caption</questionType>
      <text>This is a caption</text>
      <value/>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_2</questionId>
      <questionType>DateQuestion</questionType>
      <text>What is your favorite date?</text>
      <value/>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_3</questionId>
      <questionType>NumericValue</questionType>
      <text>What is your favorite number?</text>
      <value/>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_4</questionId>
      <questionType>RatingScale</questionType>
      <text>Rating this example:</text>
      <value/>
      <required>false</required>
      <questionChoices>
        <choice>Fair</choice>
        <choice>Poor</choice>
        <choice>Terrible</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_5</questionId>
      <questionType>ShortTextValue</questionType>
      <text>What is your favorite word?</text>
```

```
      <value/>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_6</questionId>
      <questionType>TextValue</questionType>
      <text>What is your favorite phrase?</text>
      <value/>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_7</questionId>
      <questionType>TrueFalse</questionType>
      <text>This statement is false.</text>
      <value/>
      <required>false</required>
      <questionChoices>
        <choice>True</choice>
        <choice>False</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_8</questionId>
      <questionType>LargeTextValue</questionType>
      <text>What is your favorite paragraph?</text>
      <value/>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_9</questionId>
      <questionType>YesNo</questionType>
      <text>Do you like this example?</text>
      <value/>
      <required>false</required>
      <questionChoices>
        <choice>Yes</choice>
        <choice>No</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_10</questionId>
```

```
      <questionType>Captcha</questionType>
      <text>Enter the letters that appear in the image</text>
      <value/>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl>http://example.com/Captcha</imageUrl>
      <audioUrl>http://example.com/Captcha?type=audio</audioUrl>
    </question>
    <question>
      <questionId>survey100_1_1000_11</questionId>
      <questionType>Categories</questionType>
      <text>What are your interests?</text>
      <value/>
      <required>false</required>
      <questionChoices>
        <choice>Birds</choice>
        <choice>Trains</choice>
        <choice>Sports</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_12</questionId>
      <questionType>ComboChoice</questionType>
      <text>What is your favorite color?</text>
      <value/>
      <required>false</required>
      <questionChoices>
        <choice>Cyan</choice>
        <choice>Magenta</choice>
        <choice>Yellow</choice>
        <choice>Black</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_13</questionId>
      <questionType>MultiMulti</questionType>
      <text>What primary colors would you tolerate?</text>
      <value/>
      <required>false</required>
      <questionChoices>
        <choice>Red</choice>
        <choice>Green</choice>
        <choice>Blue</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_14</questionId>
      <questionType>MultiSingle</questionType>
      <text>What is your favority primary color?</text>
      <value/>
```

```
      <required>false</required>
      <questionChoices>
        <choice>Red</choice>
        <choice>Green</choice>
        <choice>Blue</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_15</questionId>
      <questionType>MultiSingleRadio</questionType>
      <text>What is your least favorite primary color?</text>
      <value/>
      <required>false</required>
      <questionChoices>
        <choice>Red</choice>
        <choice>Green</choice>
        <choice>Blue</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_16</questionId>
      <questionType>HiddenInterests</questionType>
      <text/>
      <value/>
      <required>false</required>
      <questionChoices>
        <choice>Update cons profile</choice>
        <choice>with these interests</choice>
      </questionChoices>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_17</questionId>
      <questionType>HiddenTrueFalse</questionType>
      <text/>
      <value>True</value>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
      <audioUrl/>
    </question>
    <question>
      <questionId>survey100_1_1000_18</questionId>
      <questionType>HiddenTextValue</questionType>
      <text/>
      <value>Update the cons profile with this value</value>
      <required>false</required>
      <questionChoices/>
      <minSelected>0</minSelected>
      <maxSelected>0</maxSelected>
      <imageUrl/>
```

```
      <audioUrl/>
    </question>
  </questions>
  <recipients>
    <recipient>
      <recipientId>representative.12345</recipientId>
      <name>Senator John Q. Public</name>
      <url>http://public.senate.gov/contact.cfm</url>
      <interactive>true</interactive>
      <deliveryOptions>
        <delivery>internet</delivery>
      </deliveryOptions>
    </recipient>
    <recipient>
      <recipientId>other.12345</recipientId>
      <name>Custom Target Name</name>
      <interactive>false</interactive>
      <deliveryOptions>
        <delivery>fax</delivery>
      </deliveryOptions>
    </recipient>
  </recipients>
</getAdvocacyAlertResponse>
```

JSON response


```
{"getAdvocacyAlertResponse":{"recipients":{"recipient":
[{"recipientId":"representative.12345","interactive":"true","name":"Senator
 John Q. Public","deliveryOptions":{"delivery":"internet"},"url":"http://
public.senate.gov/contact.cfm"},
{"recipientId":"other.12345","interactive":"false","name":"Custom
 Target Name","deliveryOptions":{"delivery":"fax"}}]},"alert":
{"position":"none","internalName":"Internal
 Name","messageSubjectEditable":"optional","messageSubject":"Subject","nayCount":"0","yea
 Title","alertId":"103","interactionCount":"1000002","description":"Description","priorit
{"target":["Governor","US Senator","US Representative","State Senator","State
 Representative"]},"messageGreeting":"Dear","messageBody":"Message Body
 Text","abstainCount":"0","expire":"2013-02-03T05:18:23.680-06:00","messageBodyEditable"
 Opening Text","url":"https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&id=103","modified":"2013-02-03T05:18:23.681-06:00","messageClosing":"Si
{"content":"General","categoryId":"1"},"thumbnail":"https://
secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/103","restrictByState":{"state":
["TN","TX"]},"wordLimit":"0","issues":{"issue":
[{"content":"Issue1","issueId":"1001"},
{"content":"Issue2","issueId":"1002"}]}},"questions":{"question":
[{"audioUrl":{},"text":{},"questionId":"email","imageUrl":
{},"minSelected":"0","value":"user@example.com","questionType":"TextValue","maxSelected"
{}},{"audioUrl":{},"text":{},"questionId":"title","imageUrl":
{},"minSelected":"0","value":"Ms.","questionType":"MultiSingle","maxSelected":"0","requir
{"choice":["Mr.","Ms.","Dr."]}},{"audioUrl":{},"text":
{},"questionId":"first_name","imageUrl":
{},"minSelected":"0","value":"Jane","questionType":"TextValue","maxSelected":"0","require
{}},{"audioUrl":{},"text":{},"questionId":"last_name","imageUrl":
{},"minSelected":"0","value":"Public","questionType":"TextValue","maxSelected":"0","requi
{}},{"audioUrl":{},"text":{},"questionId":"street1","imageUrl":
{},"minSelected":"0","value":"1234 Main
 St","questionType":"TextValue","maxSelected":"0","required":"true","questionChoices":
{}},{"audioUrl":{},"text":{},"questionId":"street2","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"TextValue","maxSelected":"0","required":"false","questionChoices":
```

```
{}},{"audioUrl":{},"text":{},"questionId":"city","imageUrl":
{},"minSelected":"0","value":"AnyTown","questionType":"TextValue","maxSelected":"0","requ
{}},{"audioUrl":{},"text":{},"questionId":"state","imageUrl":
{},"minSelected":"0","value":"DC","questionType":"TextValue","maxSelected":"0","required"
{}},{"audioUrl":{},"text":{},"questionId":"zip","imageUrl":
{},"minSelected":"0","value":"20002","questionType":"TextValue","maxSelected":"0","requir
{}},{"audioUrl":{},"text":{},"questionId":"country","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"TextValue","maxSelected":"0","required":"false","questionChoices":
{}},{"audioUrl":{},"text":{},"questionId":"phone","imageUrl":
{},"minSelected":"0","value":"555-1212","questionType":"TextValue","maxSelected":"0","req
{}},{"audioUrl":{},"text":{},"questionId":"subject","imageUrl":
{},"minSelected":"0","value":"Abstain from H.R.
 12345","questionType":"TextValue","maxSelected":"0","required":"false","questionChoices"
{}},{"audioUrl":{},"text":{},"questionId":"body","imageUrl":
{},"minSelected":"0","value":"Please do not vote on this
 bill!","questionType":"TextValue","maxSelected":"0","required":"false","questionChoices"
{}},{"audioUrl":{},"text":{},"questionId":"opt_in","imageUrl":
{},"minSelected":"0","value":"true","questionType":"TrueFalse","maxSelected":"0","require
{"choice":["True","False"]}},{"audioUrl":{},"text":
{},"questionId":"anonymous","imageUrl":
{},"minSelected":"0","value":"false","questionType":"TrueFalse","maxSelected":"0","requir
{"choice":["True","False"]}},{"audioUrl":{},"text":
{},"questionId":"cc","imageUrl":
{},"minSelected":"0","value":"true","questionType":"TrueFalse","maxSelected":"0","require
{"choice":["True","False"]}},{"audioUrl":{},"text":"This is
 a caption","questionId":"survey100_1_1000_1","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"Caption","maxSelected":"0","required":"false","questionChoices":
{}},{"audioUrl":{},"text":"What is your favorite
 date?","questionId":"survey100_1_1000_2","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"DateQuestion","maxSelected":"0","required":"false","questionChoices":
{}},{"audioUrl":{},"text":"What is your favorite
 number?","questionId":"survey100_1_1000_3","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"NumericValue","maxSelected":"0","required":"false","questionChoices":
{}},{"audioUrl":{},"text":"Rating this
 example:","questionId":"survey100_1_1000_4","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"RatingScale","maxSelected":"0","required":"false","questionChoices":
{"choice":["Fair","Poor","Terrible"]}},{"audioUrl":{},"text":"What is
 your favorite word?","questionId":"survey100_1_1000_5","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"ShortTextValue","maxSelected":"0","required":"false","questionChoices"
{}},{"audioUrl":{},"text":"What is your favorite
 phrase?","questionId":"survey100_1_1000_6","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"TextValue","maxSelected":"0","required":"false","questionChoices":
{}},{"audioUrl":{},"text":"This statement is
 false.","questionId":"survey100_1_1000_7","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"TrueFalse","maxSelected":"0","required":"false","questionChoices":
{"choice":["True","False"]}},{"audioUrl":{},"text":"What is your
 favorite paragraph?","questionId":"survey100_1_1000_8","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"LargeTextValue","maxSelected":"0","required":"false","questionChoices"
{}},{"audioUrl":{},"text":"Do you like this
 example?","questionId":"survey100_1_1000_9","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"YesNo","maxSelected":"0","required":"false","questionChoices":
{"choice":["Yes","No"]}},{"audioUrl":"http://example.com/
Captcha?type=audio","text":"Enter the letters that appear in the
 image","questionId":"survey100_1_1000_10","imageUrl":"http://
```

```
example.com/Captcha","minSelected":"0","value":
{},"questionType":"Captcha","maxSelected":"0","required":"false","questionChoices":
{}},{"audioUrl":{},"text":"What are your
 interests?","questionId":"survey100_1_1000_11","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"Categories","maxSelected":"0","required":"false","questionChoices":
{"choice":["Birds","Trains","Sports"]}},{"audioUrl":{},"text":"What is
 your favorite color?","questionId":"survey100_1_1000_12","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"ComboChoice","maxSelected":"0","required":"false","questionChoices":
{"choice":["Cyan","Magenta","Yellow","Black"]}},
{"audioUrl":{},"text":"What primary colors would you
 tolerate?","questionId":"survey100_1_1000_13","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"MultiMulti","maxSelected":"0","required":"false","questionChoices":
{"choice":["Red","Green","Blue"]}},{"audioUrl":{},"text":"What is your
 favority primary color?","questionId":"survey100_1_1000_14","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"MultiSingle","maxSelected":"0","required":"false","questionChoices":
{"choice":["Red","Green","Blue"]}},{"audioUrl":{},"text":"What is your least
 favorite primary color?","questionId":"survey100_1_1000_15","imageUrl":
{},"minSelected":"0","value":
{},"questionType":"MultiSingleRadio","maxSelected":"0","required":"false","questionChoice
{"choice":["Red","Green","Blue"]}},{"audioUrl":{},"text":
{},"questionId":"survey100_1_1000_16","imageUrl":{},"minSelected":"0","value":
{},"questionType":"HiddenInterests","maxSelected":"0","required":"false","questionChoices
{"choice":["Update cons profile","with these interests"]}},
{"audioUrl":{},"text":{},"questionId":"survey100_1_1000_17","imageUrl":
{},"minSelected":"0","value":"True","questionType":"HiddenTrueFalse","maxSelected":"0","r
{}},{"audioUrl":{},"text":{},"questionId":"survey100_1_1000_18","imageUrl":
{},"minSelected":"0","value":"Update the cons profile with this
 value","questionType":"HiddenTextValue","maxSelected":"0","required":"false","questionCh
{}}]}}}
```


getAdvocacyAlerts

This method returns a list of Advocacy alerts.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAdvocacyAPI	SRAdvocacyAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAdvocacyAPI?method=getAdvocacyAlerts`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sso_auth_token=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&alert_status=`*active* | *inactive* | *archived*] [`&alert_type=`*action* | *lte* | *vote* | *call*] [`&issue_id=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*title* | *internalName* | *description* | *category* | *issues* | *restrictByState* | *targets* | *messageSubject* | *messageGreeting* | *messageOpening* | *messageBody* | *messageClosing* | *messageSignature*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*alertId* | *type* | *status* | *priority* | *url* | *interactionCount* | *title* | *internalName* | *description* | *category* | *issues* | *restrictByState* | *modified* | *publish* | *expire* | *targets* | *messageSubject* | *messageSubjectEditable* | *messageGreeting* | *messageOpening* | *messageBody* | *messageBodyEditable* | *messageClosing* | *messageSignature* | *letterLimit* | *wordLimit* | *position* | *yeaCount* | *nayCount* | *abstainCount* | *notPresentCount*] [`&match=`*value*] [`&max_expire=`*value*] [`&max_modified=`*value*] [`&max_publish=`*value*] [`&min_expire=`*value*

] [&min_modified=*value*] [&min_publish=*value*] [&priority=*low* | *medium* | *high*] [&restrict_by_state=*value*] [&taken=*ignore* | *never* | *already*]

Server API Syntax

https://*secure2.convio.net/organization*/site/SRAdvocacyAPI?method=getAdvocacyAlerts & api_key =*value* & login_name =*value* & login_password =*value* & v =*value* [& center_id =*value*] [& cons_id =*value*] [& error_redirect =*value*] [& redirect =*value*] [& response_format =*xml* |*json*] [& sign_redirects =*value*] [& source =*value*] [& sub_source =*value*] [& success_redirect =*value*] [& suppress_response_codes =*value*] [& alert_status =*active* |*inactive* |*archived*] [& alert_type =*action* |*lte* |*vote* |*call*] [& issue_id =*value*] [& list_ascending =*value*] [& list_category_id =*value*] [& list_filter_column =*title* |*internalName* |*description* |*category* |*issues* |*restrictByState* |*targets* |*messageSubject* |*messageGreeting* |*messageOpening* |*messageBody* |*messageClosing* |*messageSignature*] [& list_filter_text =*value*] [& list_page_offset =*value*] [& list_page_size =*value*] [& list_record_offset =*value*] [& list_sort_column =*alertId* |*type* |*status* |*priority* |*url* |*interactionCount* |*title* |*internalName* |*description* |*category* |*issues* |*restrictByState* |*modified* |*publish* |*expire* |*targets* |*messageSubject* |*messageSubjectEditable* |*messageGreeting* |*messageOpening* |*messageBody* |*messageBodyEditable* |*messageClosing* |*messageSignature* |*letterLimit* |*wordLimit* |*position* |*yeaCount* |*nayCount* |*abstainCount* |*notPresentCount*] [& match =*value*] [& max_expire =*value*] [& max_modified =*value*] [& max_publish =*value*] [& min_expire =*value*] [& min_modified =*value*] [& min_publish =*value*] [& priority =*low* |*medium* |*high*] [& restrict_by_state =*value*] [& taken =*ignore* |*never* |*already*]

Usage Notes

Call `getAdvocacyAlerts` to obtain a list of advocacy alerts and information about each alert. Optionally, specify additional parameters to limit and order the result set.

Filtering, Sorting and Paging Results

To control which alerts are returned, specify filter parameters. Available filters include:

- alert_status
- alert_type
- issue_id
- match
- priority
- restrict_by_state
- taken

Optionally specify publication, expiration, or last modification date and time limits or ranges using min and max parameters. Note that these fields and the ranges they define include date, time and time zone offset components in ISO 8601 format. You specify both date and time when publishing an alert. The time zone offset applied is the one configured for your Convio site.

Note that the taken and match filters apply to alerts which are associated with an individual constituent, either by cons_id parameter or through the caller's current logged-in user session.

In order to filter alerts associated with a given issue, you must specify one or more issue_id values. To locate the issue_id value for a particular issue in Convio, select Issues from the Convio Advocacy menu. The ID for each issue appears in parentheses beside its the name in the list.

You can further filter the list of results using the list_filter_column and list_filter_text parameters.

To sort the results, specify list_sort_column and list_ascending parameters.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

alert_status

Optional. Filter by alert status. Accepts a comma separated list. Active alerts allow qualified participants to take action, inactive and archived alerts do not. Inactive alerts have been active at

some point but no longer are, typically because they have expired, but they have not been archived by an administrator.

Type advocacyAlertStatus.

Default is active.

Options are:

- active
- inactive
- archived

alert_type

Optional. Filter by alert type. Accepts a comma separated list. The default is 'action,lte,vote,call' (all types).

Type advocacyAlertType.

Options are:

- action
- lte
- vote
- call

issue_id

Optional. Filter by alert issues. Accepts a comma separated list of valid Issue IDs. The default is an empty list, which causes this filter to be ignored. Only alerts associated with all of the specified issues will be returned.

Type xsd:positiveInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

Options are:

- title
- internalName
- description
- category
- issues
- restrictByState
- targets
- messageSubject
- messageGreeting
- messageOpening
- messageBody

- messageClosing
- messageSignature

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Options are:

- alertId
- type
- status
- priority
- url
- interactionCount
- title
- internalName
- description
- category
- issues
- restrictByState
- modified
- publish
- expire
- targets
- messageSubject
- messageSubjectEditable
- messageGreeting
- messageOpening
- messageBody
- messageBodyEditable
- messageClosing
- messageSignature
- letterLimit

- wordLimit
- position
- yeaCount
- nayCount
- abstainCount
- notPresentCount

match

Optional. Filter by geographic match. If true, returns only alerts where the constituent's address matches a target recipient for the alert. The default is 'false'.

Type xsd:boolean.

Default is false.

max_expire

Optional. Filter by expiration timestamp. If you specify a timestamp, only alerts expiring on or before the specified timestamp will be returned.

Type xsd:dateTime.

max_modified

Optional. Filter by last modified timestamp. If you specify a timestamp, only alerts last modified on or before the specified timestamp will be returned.

Type xsd:dateTime.

max_publish

Optional. Filter by publish timestamp. If you specify a timestamp, only alerts published on or before the specified timstamp will be returned.

Type xsd:dateTime.

min_expire

Optional. Filter by expiration timestamp. If you specify a timestamp, only alerts expiring on or after the specified timestamp will be returned.

Type xsd:dateTime.

min_modified

Optional. Filter by last modified timestamp. If you specify a timestamp, only alerts last modified on or after the specified timestamp will be returned.

Type xsd:dateTime.

min_publish

Optional. Filter by publish timestamp. If you specify a timestamp, only alerts published on or after the specified timestamp will be returned.

Type xsd:dateTime.

priority

Optional. Filter by alert priority. Accepts a comma separated list. The default is 'high,medium,low' (all priorities).

Type advocacyPriority.

Options are:

- low
- medium
- high

restrict_by_state

Optional. Filter by alert "restrict by state" setting. Accepts a comma separated list of US Postal abbreviations. The default is an empty list, which causes this filter to be ignored. Only alerts that list all of the specified abbreviations will be returned.

Type xsd:string.

taken

Optional. For alerts that apply to a given constituent, filter by whether the constituent has taken action on the alert. The default is 'ignore'.

Type advocacyTaken.

Default is ignore.

Options are:

- ignore
- never
- already

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Optional. Specifies the constituent associated with the alert or action.

Type xsd:nonNegativeInteger.

sign_redirects

Optional. Specifies that redirect URLs should be signed with a timestamp and a hash of the query string arguments and a secret key.

Type xsd:boolean.

Default is false.

alert_status

Optional. Filter by alert status. Accepts a comma separated list. Active alerts allow qualified participants to take action, inactive and archived alerts do not. Inactive alerts have been active at some point but no longer are, typically because they have expired, but they have not been archived by an administrator.

Type advocacyAlertStatus.

Default is active.

Options are:

- active
- inactive
- archived

alert_type

Optional. Filter by alert type. Accepts a comma separated list. The default is 'action,lte,vote,call' (all types).

Type advocacyAlertType.

Options are:

- action

- lte
- vote
- call

issue_id

Optional. Filter by alert issues. Accepts a comma separated list of valid Issue IDs. The default is an empty list, which causes this filter to be ignored. Only alerts associated with all of the specified issues will be returned.

Type xsd:positiveInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

Options are:

- title
- internalName
- description
- category
- issues
- restrictByState
- targets
- messageSubject
- messageGreeting
- messageOpening
- messageBody
- messageClosing
- messageSignature

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Options are:

- alertId
- type
- status
- priority
- url
- interactionCount
- title
- internalName
- description
- category
- issues
- restrictByState
- modified
- publish
- expire
- targets
- messageSubject
- messageSubjectEditable
- messageGreeting
- messageOpening
- messageBody
- messageBodyEditable
- messageClosing
- messageSignature
- letterLimit
- wordLimit
- position
- yeaCount
- nayCount
- abstainCount
- notPresentCount

match

Optional. Filter by geographic match. If true, returns only alerts where the constituent's address matches a target recipient for the alert. The default is 'false'.

Type xsd:boolean.

Default is false.

max_expire

Optional. Filter by expiration timestamp. If you specify a timestamp, only alerts expiring on or before the specified timestamp will be returned.

Type xsd:dateTime.

max_modified

Optional. Filter by last modified timestamp. If you specify a timestamp, only alerts last modified on or before the specified timestamp will be returned.

Type xsd:dateTime.

max_publish

Optional. Filter by publish timestamp. If you specify a timestamp, only alerts published on or before the specified timstamp will be returned.

Type xsd:dateTime.

min_expire

Optional. Filter by expiration timestamp. If you specify a timestamp, only alerts expiring on or after the specified timestamp will be returned.

Type xsd:dateTime.

min_modified

Optional. Filter by last modified timestamp. If you specify a timestamp, only alerts last modified on or after the specified timestamp will be returned.

Type xsd:dateTime.

min_publish

Optional. Filter by publish timestamp. If you specify a timestamp, only alerts published on or after the specified timestamp will be returned.

Type xsd:dateTime.

priority

Optional. Filter by alert priority. Accepts a comma separated list. The default is 'high,medium,low' (all priorities).

Type advocacyPriority.

Options are:

- low
- medium
- high

restrict_by_state

Optional. Filter by alert "restrict by state" setting. Accepts a comma separated list of US Postal abbreviations. The default is an empty list, which causes this filter to be ignored. Only alerts that list all of the specified abbreviations will be returned.

Type xsd:string.

taken

Optional. For alerts that apply to a given constituent, filter by whether the constituent has taken action on the alert. The default is 'ignore'.

Type advocacyTaken.

Default is ignore.

Options are:

- ignore
- never

- already

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getAdvocacyAlertsResponse alertCount="20" xsi:schemaLocation="http://
convio.com/crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd"
 xmlns="http://convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <alert>
    <alertId>103</alertId>
    <type>action</type>
    <status>active</status>
    <priority>high</priority>
    <url>https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&amp;id=103</url>
    <interactionCount>1000002</interactionCount>
    <title>Alert Title</title>
    <thumbnail>https://secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/103</thumbnail>
    <internalName>Internal Name</internalName>
    <description>Description</description>
    <category categoryId="1">General</category>
    <issues>
      <issue issueId="1001">Issue1</issue>
      <issue issueId="1002">Issue2</issue>
    </issues>
    <restrictByState>
      <state>TN</state>
      <state>TX</state>
    </restrictByState>
    <modified>2013-02-03T05:18:23.340-06:00</modified>
    <publish>2013-02-03T05:18:23.340-06:00</publish>
    <expire>2013-02-03T05:18:23.339-06:00</expire>
    <targets>
      <target>Governor</target>
      <target>US Senator</target>
      <target>US Representative</target>
      <target>State Senator</target>
      <target>State Representative</target>
    </targets>
    <messageSubject>Subject</messageSubject>
    <messageSubjectEditable>optional</messageSubjectEditable>
    <messageGreeting>Dear</messageGreeting>
    <messageOpening>Message Opening Text</messageOpening>
    <messageBody>Message Body Text</messageBody>
    <messageBodyEditable>optional</messageBodyEditable>
    <messageClosing>Sincerely,</messageClosing>
    <messageSignature>Signature</messageSignature>
    <wordLimit>0</wordLimit>
    <position>none</position>
    <yeaCount>0</yeaCount>
    <nayCount>0</nayCount>
    <abstainCount>0</abstainCount>
  </alert>
  <alert>
    <alertId>116</alertId>
    <type>action</type>
    <status>active</status>
    <priority>high</priority>
```

```
    <url>https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&amp;id=116</url>
    <interactionCount>1000002</interactionCount>
    <title>Alert Title</title>
    <thumbnail>https://secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/116</thumbnail>
    <internalName>Internal Name</internalName>
    <description>Description</description>
    <category categoryId="1">General</category>
    <issues>
      <issue issueId="1001">Issue1</issue>
      <issue issueId="1002">Issue2</issue>
    </issues>
    <restrictByState>
      <state>TN</state>
      <state>TX</state>
    </restrictByState>
    <modified>2013-02-03T05:18:23.343-06:00</modified>
    <publish>2013-02-03T05:18:23.343-06:00</publish>
    <expire>2013-02-03T05:18:23.342-06:00</expire>
    <targets>
      <target>Governor</target>
      <target>US Senator</target>
      <target>US Representative</target>
      <target>State Senator</target>
      <target>State Representative</target>
    </targets>
    <messageSubject>Subject</messageSubject>
    <messageSubjectEditable>optional</messageSubjectEditable>
    <messageGreeting>Dear</messageGreeting>
    <messageOpening>Message Opening Text</messageOpening>
    <messageBody>Message Body Text</messageBody>
    <messageBodyEditable>optional</messageBodyEditable>
    <messageClosing>Sincerely,</messageClosing>
    <messageSignature>Signature</messageSignature>
    <wordLimit>0</wordLimit>
    <position>none</position>
    <yeaCount>0</yeaCount>
    <nayCount>0</nayCount>
    <abstainCount>0</abstainCount>
  </alert>
</getAdvocacyAlertsResponse>
```


JSON response


```
{"getAdvocacyAlertsResponse":{"alert":
[{"position":"none","internalName":"Internal
 Name","messageSubjectEditable":"optional","messageSubject":"Subject","nayCount":"0","yea
 Title","alertId":"103","interactionCount":"1000002","description":"Description","priorit
{"target":["Governor","US Senator","US Representative","State Senator","State
 Representative"]},"messageGreeting":"Dear","messageBody":"Message Body
 Text","abstainCount":"0","expire":"2013-02-03T05:18:23.384-06:00","messageBodyEditable"
 Opening Text","url":"https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&id=103","modified":"2013-02-03T05:18:23.385-06:00","messageClosing":"Si
{"content":"General","categoryId":"1"},"thumbnail":"https://
secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/103","restrictByState":{"state":
["TN","TX"]},"wordLimit":"0","issues":{"issue":
[{"content":"Issue1","issueId":"1001"},
{"content":"Issue2","issueId":"1002"}]}},
{"position":"none","internalName":"Internal
```

```
 Name","messageSubjectEditable":"optional","messageSubject":"Subject","nayCount":"0","yea
 Title","alertId":"116","interactionCount":"1000002","description":"Description","priorit
{"target":["Governor","US Senator","US Representative","State Senator","State
 Representative"]},"messageGreeting":"Dear","messageBody":"Message Body
 Text","abstainCount":"0","expire":"2013-02-03T05:18:23.386-06:00","messageBodyEditable"
 Opening Text","url":"https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&id=116","modified":"2013-02-03T05:18:23.387-06:00","messageClosing":"Si
{"content":"General","categoryId":"1"},"thumbnail":"https://
secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/116","restrictByState":{"state":
["TN","TX"]},"wordLimit":"0","issues":{"issue":
[{"content":"Issue1","issueId":"1001"},
{"content":"Issue2","issueId":"1002"}]}}],"alertCount":"20"}}
```


getAdvocacyInteractions

This method returns a list of Advocacy interactions by a constituent.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAdvocacyAPI	SRAdvocacyAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAdvocacyAPI?method=getAdvocacyInteractions &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sso_auth_token=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&alert_status=`*active* | *inactive* | *archived*] [`&alert_type=`*action* | *lte* | *vote* | *call*] [`&issue_id=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*title* | *internalName* | *description* | *category* | *issues* | *restrictByState* | *targets* | *messageSubject* | *messageGreeting* | *messageOpening* | *messageBody* | *messageClosing* | *messageSignature* | *subject* | *body* | *contactPosition* | *contactName* | *reply* | *note*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*alertId* | *type* | *status* | *priority* | *url* | *interactionCount* | *title* | *internalName* | *description* | *category* | *issues* | *restrictByState* | *modified* | *publish* | *expire* | *targets* | *messageSubject* | *messageSubjectEditable* | *messageGreeting* | *messageOpening* | *messageBody* | *messageBodyEditable* | *messageClosing* | *messageSignature* | *letterLimit* | *wordLimit* | *position* | *yeaCount* | *nayCount* | *abstainCount* | *notPresentCount* | *interactionId* | *interacted* | *alert* | *subject* | *body* | *called* | *contactPosition* | *contactName* | *reply* | *note*] [`&max_expire=`*value*] [`&max_interacted=`*value*] [`&max_modified=`*value*] [`&max_publish=`*value*] [`&min_expire=`*value*] [`&min_interacted=`*value*] [`&min_modified=`*value*] [`&min_publish=`*value*] [`&priority=`*low* | *medium* | *high*] [`&restrict_by_state=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRAdvocacyAPI?method=getAdvocacyInteractions &api_key=`*value* `&cons_id=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&alert_status=`*active* | *inactive* | *archived*] [`&alert_type=`*action* | *lte* | *vote* | *call*] [`&issue_id=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*title* | *internalName* | *description* | *category* | *issues* | *restrictByState* | *targets* | *messageSubject* | *messageGreeting* | *messageOpening* | *messageBody* | *messageClosing* | *messageSignature* | *subject* | *body* | *contactPosition* | *contactName* | *reply* | *note*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*alertId* | *type* | *status* | *priority* | *url* | *interactionCount* | *title* | *internalName* | *description* | *category* | *issues* | *restrictByState* | *modified* | *publish*

|*expire* |*targets* |*messageSubject* |*messageSubjectEditable* |*messageGreeting* |*messageOpening* |*messageBody* |*messageBodyEditable* |*messageClosing* |*messageSignature* |*letterLimit* |*wordLimit* |*position* |*yeaCount* |*nayCount* |*abstainCount* |*notPresentCount* |*interactionId* |*interacted* |*alert* |*subject* |*body* |*called* |*contactPosition* |*contactName* |*reply* |*note*] [& `max_expire` =*value*] [& `max_interacted` =*value*] [& `max_modified` =*value*] [& `max_publish` =*value*] [& `min_expire` =*value*] [& `min_interacted` =*value*] [& `min_modified` =*value*] [& `min_publish` =*value*] [& `priority` =*low* |*medium* |*high*] [& `restrict_by_state` =*value*]

Usage Notes

Call getAdvocacyInteractions to obtain advocacy interactions for a specific constituent. Optionally, specify additional parameters to limit the result set.

Filtering, Sorting and Paging Results

To control which alerts are returned, specify filter parameters. Available filters include:

- alert_status
- alert_type
- issue_id
- priority
- restrict_by_state

Though these filters apply to attributes of the underlying alerts, not to interactions, you can use them to narrow the list of interactions to actions taken on alerts with specific attributes.

Optionally specify date and time limits or ranges for interaction, publication, expiration or last modification using min and max parameters. The publication, expiration, and last modification dates refer to action alerts. The interacted date is the date on which the calling constituent took action on each alert. Values for these fields and the ranges they define include date, time and time zone offset information in ISO 8601 format. You specify both the date and time when publishing an alert. The time zone offset applied is the one configured for your Convio site.

In order to filter alerts associated with a given issue, you must specify one or more issue_id values. To locate the issue_id value for a particular issue in Convio, select Issues from the Convio Advocacy menu. The ID for each issue appears in parentheses beside its name in the list.

You can further filter the list of results using the list_filter_column and list_filter_text parameters.

To sort the results, specify list_sort_column and list_ascending parameters.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

alert_status

Optional. Filter by alert status. Accepts a comma separated list. Active alerts allow qualified participants to take action, inactive and archived alerts do not. Inactive alerts have been active at some point but no longer are, typically because they have expired, but they have not been archived by an administrator. The default is 'active,inactive,archived' (all statuses).

Type advocacyAlertStatus.

Options are:

- active
- inactive
- archived

alert_type

Optional. Filter by alert type. Accepts a comma separated list. The default is 'action,lte,vote,call' (all types).

Type advocacyAlertType.

Options are:

- action
- lte
- vote
- call

issue_id

Optional. Filter by alert issues. Accepts a comma separated list of valid Issue IDs. The default is an empty list, which causes this filter to be ignored. Only alerts associated with all of the specified issues will be returned.

Type xsd:positiveInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

Options are:

- title
- internalName
- description
- category
- issues
- restrictByState
- targets
- messageSubject
- messageGreeting
- messageOpening
- messageBody
- messageClosing
- messageSignature
- subject
- body
- contactPosition
- contactName
- reply
- note

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Options are:

- alertId
- type
- status
- priority
- url
- interactionCount
- title
- internalName
- description
- category
- issues
- restrictByState
- modified
- publish
- expire
- targets
- messageSubject
- messageSubjectEditable
- messageGreeting
- messageOpening
- messageBody
- messageBodyEditable
- messageClosing
- messageSignature
- letterLimit
- wordLimit
- position
- yeaCount
- nayCount
- abstainCount
- notPresentCount
- interactionId
- interacted

- alert
- subject
- body
- called
- contactPosition
- contactName
- reply
- note

max_expire

Optional. Filter by expiration timestamp. If you specify a timestamp, only alerts expiring on or before the specified timestamp will be returned.

Type xsd:dateTime.

max_interacted

Optional. Filter by interaction timestamp. If you specify a timestamp, only interactions on or before the specified timestamp will be returned.

Type xsd:dateTime.

max_modified

Optional. Filter by last modified timestamp. If you specify a timestamp, only alerts last modified on or before the specified timestamp will be returned.

Type xsd:dateTime.

max_publish

Optional. Filter by publish timestamp. If you specify a timestamp, only alerts published on or before the specified timstamp will be returned.

Type xsd:dateTime.

min_expire

Optional. Filter by expiration timestamp. If you specify a timestamp, only alerts expiring on or after the specified timestamp will be returned.

Type xsd:dateTime.

min_interacted

Optional. Filter by interaction timestamp. If you specify a timestamp, only interactions on or after the specified timestamp will be returned.

Type xsd:dateTime.

min_modified

Optional. Filter by last modified timestamp. If you specify a timestamp, only alerts last modified on or after the specified timestamp will be returned.

Type xsd:dateTime.

min_publish

Optional. Filter by publish timestamp. If you specify a timestamp, only alerts published on or after the specified timestamp will be returned.

Type xsd:dateTime.

priority

Optional. Filter by alert priority. Accepts a comma separated list. The default is 'high,medium,low' (all priorities).

Type advocacyPriority.

Options are:

- low
- medium
- high

restrict_by_state

Optional. Filter by alert "restrict by state" setting. Accepts a comma separated list of US Postal abbreviations. The default is an empty list, which causes this filter to be ignored. Only alerts that list all of the specified abbreviations will be returned.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Required. Specifies the constituent associated with the alert or action.

Type xsd:nonNegativeInteger.

sign_redirects

Optional. Specifies that redirect URLs should be signed with a timestamp and a hash of the query string arguments and a secret key.

Type xsd:boolean.

Default is false.

alert_status

Optional. Filter by alert status. Accepts a comma separated list. Active alerts allow qualified participants to take action, inactive and archived alerts do not. Inactive alerts have been active at some point but no longer are, typically because they have expired, but they have not been archived by an administrator. The default is 'active,inactive,archived' (all statuses).

Type advocacyAlertStatus.

Options are:

- active
- inactive
- archived

alert_type

Optional. Filter by alert type. Accepts a comma separated list. The default is 'action,lte,vote,call' (all types).

Type advocacyAlertType.

Options are:

- action
- lte
- vote
- call

issue_id

Optional. Filter by alert issues. Accepts a comma separated list of valid Issue IDs. The default is an empty list, which causes this filter to be ignored. Only alerts associated with all of the specified issues will be returned.

Type xsd:positiveInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

Options are:

- title
- internalName
- description
- category
- issues
- restrictByState
- targets
- messageSubject
- messageGreeting
- messageOpening
- messageBody
- messageClosing
- messageSignature
- subject
- body
- contactPosition
- contactName
- reply
- note

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Options are:

- alertId
- type
- status
- priority
- url
- interactionCount
- title
- internalName
- description
- category
- issues
- restrictByState
- modified
- publish
- expire
- targets
- messageSubject
- messageSubjectEditable
- messageGreeting
- messageOpening
- messageBody
- messageBodyEditable
- messageClosing
- messageSignature
- letterLimit
- wordLimit
- position
- yeaCount
- nayCount
- abstainCount
- notPresentCount
- interactionId
- interacted
- alert
- subject
- body
- called
- contactPosition
- contactName
- reply
- note

max_expire

Optional. Filter by expiration timestamp. If you specify a timestamp, only alerts expiring on or before the specified timestamp will be returned.

Type xsd:dateTime.

max_interacted

Optional. Filter by interaction timestamp. If you specify a timestamp, only interactions on or before the specified timestamp will be returned.

Type xsd:dateTime.

max_modified

Optional. Filter by last modified timestamp. If you specify a timestamp, only alerts last modified on or before the specified timestamp will be returned.

Type xsd:dateTime.

max_publish

Optional. Filter by publish timestamp. If you specify a timestamp, only alerts published on or before the specified timstamp will be returned.

Type xsd:dateTime.

min_expire

Optional. Filter by expiration timestamp. If you specify a timestamp, only alerts expiring on or after the specified timestamp will be returned.

Type xsd:dateTime.

min_interacted

Optional. Filter by interaction timestamp. If you specify a timestamp, only interactions on or after the specified timestamp will be returned.

Type xsd:dateTime.

min_modified

Optional. Filter by last modified timestamp. If you specify a timestamp, only alerts last modified on or after the specified timestamp will be returned.

Type xsd:dateTime.

min_publish

Optional. Filter by publish timestamp. If you specify a timestamp, only alerts published on or after the specified timestamp will be returned.

Type xsd:dateTime.

priority

Optional. Filter by alert priority. Accepts a comma separated list. The default is 'high,medium,low' (all priorities).

Type advocacyPriority.

Options are:
- low
- medium
- high

restrict_by_state

Optional. Filter by alert "restrict by state" setting. Accepts a comma separated list of US Postal abbreviations. The default is an empty list, which causes this filter to be ignored. Only alerts that list all of the specified abbreviations will be returned.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getAdvocacyInteractionsResponse interactionCount="30"
 xsi:schemaLocation="http://convio.com/crm/v1.0 http://service.convio.net/
xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/v1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <interaction>
    <interactionId>1001</interactionId>
    <interacted>2013-02-03T05:18:23.596-06:00</interacted>
    <alert>
      <alertId>103</alertId>
      <type>action</type>
      <status>active</status>
      <priority>high</priority>
      <url>https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&amp;id=103</url>
      <interactionCount>1000002</interactionCount>
      <title>Alert Title</title>
      <thumbnail>https://secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/103</thumbnail>
      <internalName>Internal Name</internalName>
      <description>Description</description>
      <category categoryId="1">General</category>
      <issues>
        <issue issueId="1001">Issue1</issue>
        <issue issueId="1002">Issue2</issue>
      </issues>
      <restrictByState>
        <state>TN</state>
        <state>TX</state>
      </restrictByState>
      <modified>2013-02-03T05:18:23.597-06:00</modified>
      <publish>2013-02-03T05:18:23.597-06:00</publish>
      <expire>2013-02-03T05:18:23.596-06:00</expire>
      <targets>
        <target>Governor</target>
        <target>US Senator</target>
        <target>US Representative</target>
        <target>State Senator</target>
        <target>State Representative</target>
      </targets>
      <messageSubject>Subject</messageSubject>
      <messageSubjectEditable>optional</messageSubjectEditable>
      <messageGreeting>Dear</messageGreeting>
      <messageOpening>Message Opening Text</messageOpening>
      <messageBody>Message Body Text</messageBody>
      <messageBodyEditable>optional</messageBodyEditable>
      <messageClosing>Sincerely,</messageClosing>
      <messageSignature>Signature</messageSignature>
      <wordLimit>0</wordLimit>
      <position>none</position>
      <yeaCount>0</yeaCount>
      <nayCount>0</nayCount>
      <abstainCount>0</abstainCount>
    </alert>
    <recipients>
      <recipient>
```

```
        <recipientId>ID</recipientId>
        <name>Senator John Q. Public</name>
        <url>http://public.senate.gov/contact.cfm</url>
        <interactive>true</interactive>
        <deliveryOptions>
          <delivery>internet</delivery>
        </deliveryOptions>
      </recipient>
    </recipients>
    <subject>Subject</subject>
    <body>Body</body>
    <called>2013-02-03T05:18:23.596-06:00</called>
    <contactPosition>ContactPosition</contactPosition>
    <contactName>ContactName</contactName>
    <reply>Reply</reply>
    <note>Note</note>
  </interaction>
  <interaction>
    <interactionId>1002</interactionId>
    <interacted>2013-02-03T05:18:23.597-06:00</interacted>
    <alert>
      <alertId>103</alertId>
      <type>action</type>
      <status>active</status>
      <priority>high</priority>
      <url>https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&amp;id=103</url>
      <interactionCount>1000002</interactionCount>
      <title>Alert Title</title>
      <thumbnail>https://secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/103</thumbnail>
      <internalName>Internal Name</internalName>
      <description>Description</description>
      <category categoryId="1">General</category>
      <issues>
        <issue issueId="1001">Issue1</issue>
        <issue issueId="1002">Issue2</issue>
      </issues>
      <restrictByState>
        <state>TN</state>
        <state>TX</state>
      </restrictByState>
      <modified>2013-02-03T05:18:23.598-06:00</modified>
      <publish>2013-02-03T05:18:23.598-06:00</publish>
      <expire>2013-02-03T05:18:23.598-06:00</expire>
      <targets>
        <target>Governor</target>
        <target>US Senator</target>
        <target>US Representative</target>
        <target>State Senator</target>
        <target>State Representative</target>
      </targets>
      <messageSubject>Subject</messageSubject>
      <messageSubjectEditable>optional</messageSubjectEditable>
      <messageGreeting>Dear</messageGreeting>
      <messageOpening>Message Opening Text</messageOpening>
      <messageBody>Message Body Text</messageBody>
      <messageBodyEditable>optional</messageBodyEditable>
      <messageClosing>Sincerely,</messageClosing>
      <messageSignature>Signature</messageSignature>
      <wordLimit>0</wordLimit>
      <position>none</position>
      <yeaCount>0</yeaCount>
      <nayCount>0</nayCount>
```

```
      <abstainCount>0</abstainCount>
    </alert>
    <recipients>
      <recipient>
        <recipientId>ID</recipientId>
        <name>Senator John Q. Public</name>
        <url>http://public.senate.gov/contact.cfm</url>
        <interactive>true</interactive>
        <deliveryOptions>
          <delivery>internet</delivery>
        </deliveryOptions>
      </recipient>
    </recipients>
    <subject>Subject</subject>
    <body>Body</body>
    <called>2013-02-03T05:18:23.597-06:00</called>
    <contactPosition>ContactPosition</contactPosition>
    <contactName>ContactName</contactName>
    <reply>Reply</reply>
    <note>Note</note>
  </interaction>
</getAdvocacyInteractionsResponse>
```

JSON response


```
{"getAdvocacyInteractionsResponse":{"interactionCount":"30","interaction":
[{"body":"Body","interactionId":"1001","recipients":{"recipient":
{"recipientId":"ID","interactive":"true","name":"Senator John Q.
 Public","deliveryOptions":{"delivery":"internet"},"url":"http://
public.senate.gov/
contact.cfm"}},"contactName":"ContactName","subject":"Subject","alert":
{"position":"none","internalName":"Internal
 Name","messageSubjectEditable":"optional","messageSubject":"Subject","nayCount":"0","yea
 Title","alertId":"103","interactionCount":"1000002","description":"Description","priorit
{"target":["Governor","US Senator","US Representative","State Senator","State
 Representative"]},"messageGreeting":"Dear","messageBody":"Message Body
 Text","abstainCount":"0","expire":"2013-02-03T05:18:23.616-06:00","messageBodyEditable"
 Opening Text","url":"https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&id=103","modified":"2013-02-03T05:18:23.616-06:00","messageClosing":"Si
{"content":"General","categoryId":"1"},"thumbnail":"https://
secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/103","restrictByState":{"state":
["TN","TX"]},"wordLimit":"0","issues":{"issue":
[{"content":"Issue1","issueId":"1001"},
{"content":"Issue2","issueId":"1002"}]}},"called":"2013-02-03T05:18:23.615-06:00","reply"
{"body":"Body","interactionId":"1002","recipients":{"recipient":
{"recipientId":"ID","interactive":"true","name":"Senator John Q.
 Public","deliveryOptions":{"delivery":"internet"},"url":"http://
public.senate.gov/
contact.cfm"}},"contactName":"ContactName","subject":"Subject","alert":
{"position":"none","internalName":"Internal
 Name","messageSubjectEditable":"optional","messageSubject":"Subject","nayCount":"0","yea
 Title","alertId":"103","interactionCount":"1000002","description":"Description","priorit
{"target":["Governor","US Senator","US Representative","State Senator","State
 Representative"]},"messageGreeting":"Dear","messageBody":"Message Body
 Text","abstainCount":"0","expire":"2013-02-03T05:18:23.617-06:00","messageBodyEditable"
 Opening Text","url":"https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&id=103","modified":"2013-02-03T05:18:23.617-06:00","messageClosing":"Si
{"content":"General","categoryId":"1"},"thumbnail":"https://
secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/103","restrictByState":{"state":
```

```
["TN","TX"]},"wordLimit":"0","issues":{"issue":
[{"content":"Issue1","issueId":"1001"},
{"content":"Issue2","issueId":"1002"}]}},"called":"2013-02-03T05:18:23.616-06:00","reply'
```


Examples

```
http://example.com/site/SRAdvocacyAPI?
method=getAdvocacyInteractions&cons_id=10001001&api_key=exampleKey&v=1.0&login_name=examp
```

takeAction

This method takes action on an alert, dispatching messages to recipients and recording interaction data.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRAdvocacyAPI	SRAdvocacyAPI	POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRAdvocacyAPI?method=takeAction` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sso_auth_token=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&alert_id=`*value* `&alert_type=`*action* | *call* | *lte* [`&anonymous=`*value*] [`&body=`*value*] [`&called=`*value*] [`&cc=`*value*] [`&city=`*value*] [`&contact_name=`*value*] [`&contact_position=`*value*] [`&country=`*value*] [`&email=`*value*] [`&first_name=`*value*] [`&last_name=`*value*] [`&note=`*value*] [`&opt_in=`*value*] [`&phone=`*value*] [`&preview=`*value*] [`&recipient=`*all* | *internet* | *fax* | *print* | *call*] [`&reply=`*value*] [`&state=`*value*] [`&street1=`*value*] [`&street2=`*value*] [`&subject=`*value*] [`&survey=`*value*] [`&title=`*value*] [`&zip=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRAdvocacyAPI?method=takeAction` `&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&cons_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&alert_id=`*value* `&alert_type=`*action* | *call* | *lte* [`&anonymous=`*value*] [`&body=`*value*] [`&called=`*value*] [`&cc=`*value*] [`&city=`*value*] [`&contact_name=`*value*] [`&contact_position=`*value*] [`&country=`*value*] [`&email=`*value*] [`&first_name=`*value*] [`&last_name=`*value*] [`&note=`*value*] [`&opt_in=`*value*] [`&phone=`*value*] [`&preview=`*value*] [`&recipient=`*all* | *internet* | *fax* | *print* | *call*] [`&reply=`*value*] [`&state=`*value*] [`&street1=`*value*] [`&street2=`*value*] [`&subject=`*value*] [`&survey=`*value*] [`&title=`*value*] [`&zip=`*value*]

Usage Notes

VERY IMPORTANT: Testing with the Preview Option

Always set the preview parameter *true* when developing and testing with the `takeAction` API. This prevents the system from sending out unwanted messages and logging alert participation interactions in the database, but behaves just like a live interaction in other respects, returning the same API response that the live interaction would return.

General Usage

Use the `takeAction` method to respond to an Advocacy alert. You must specify both the alert_id and the alert_type of the alert you wish to retrieve, as alert_id is not guaranteed unique across types.

Input parameters for `takeAction` may also require specific information returned by *getAdvocacyAlert* such as predefined message text, constituent address information, survey questions, and valid recipient IDs. When the valid

recipient list varies depending on the consitutuent's legislative district, `getAdvocacyAlert` method determines and returns the correct list based on the address of record for the logged-in user, or using an address provided as input.

Constituent Information

Many action alerts require constituent name and address information. If the constituent is logged in (or if the constituent record can be uniquely identified and the Convio system is configured to allow it), the passed-in values of street1, street2, city, state, zip, country, and phone parameters will be used to update the constituent profile.

If a unique constituent record cannot be identified, the first_name, last_name, and email parameters along with the address and phone information will be used to create a new constituent profile.

Recipients

Specify recipients and delivery methods for the alert message by passing recipient parameters. To create recipient parameters, append the `<recipientId>` element value returned by `getAdvocacyAlert` to the word "recipient" as the parameter name, and assign a delivery method as the parameter value. The valid delivery method options for each recipient appear in the `<deliveryOptions>` element returned by `getAdvocacyAlert`. For example, the parameter value pair to send an alert response to the Office of the President of the United States via all defined delivery methods is:

```
recipientrepresentative.-1=all
```

By default, if you do not specify recipients when calling `takeAction` on an action alert, the system will attempt delivery to all the recipients listed by `getAdvocacyAlert` for the action alert, using the delivery method(s) you defined when you created the alert.

Call alerts and LTE alerts require you to specify at least one valid recipient.

Survey Questions

If the action alert includes a survey, the survey question identifiers used to create survey response parameters are returned in the `questionId` element of the `getAdvocacyAlert` response for each survey question, for example: `<questionId>survey107_1040_1_1060</questionId>`.

Where the `<questionType>` of the survey question is one of the following:

- `Categories`
- `HiddenInterests`
- `HiddenTextValue`
- `HiddenTrueFalse`
- `LargeTextValue`
- `MultiMulti`
- `MultiSingle`
- `MultiSingleRadio`
- `NumericValue`
- `RatingScale`
- `ShortTextValue`
- `TextValue`
- `TrueFalse`
- `YesNo`

Returning survey data requires passing the `<questionId>` value as the parameter name and the response as the parameter value to `takeAction`. For example, the form input for a `MultiSingle` type question where the `<questionId>` element value is `survey107_1040_1_1060` might be:

```
<p>How do you normally commute to work?</p>
<select name="survey107_1040_1_1060">
  <option value="Walk, Run or Bike">Walk, Run or Bike</option>
  <option value="Mass Transit">Mass Transit</option>
  <option value="Carpool">Carpool</option>
  <option value="Other">Other</option>
</select>
```

You can respond to `Captcha` survey questions through the client API in the same way, using the `questionId` node value as the parameter name.

Note: Avoid using a `Captcha` survey question on alerts processed through the server API. To succeed, the download of the CAPTCHA test and the response must originate from the same session; the user's browser session where the image or audio file is rendered will not match the server session from which the API is called. When using the server API, it is better to implement CAPTCHA on the local server than to use a survey question on the alert.

Survey questions with a `<questionType>` of `DateQuestion` or `ComboChoice` are hadled differently from other types. For information on handling these types refer to the *Examples* section below.

Interactive Web Forms

Web-based contact forms sometimes require users to complete an interactive challenge/response test (also known as CAPTCHA) to limit spam. The `takeAction` API cannot deliver messages via an interactive web form. In particular, a significant percentage (150-200 out of 535) of Federal congressional webforms are interactive. Alternatively, you can direct users to the recipient's contact web site so they can interact with the web form directly. The `<url>` element returned by `getAdvocacyAlert` contains the link to the recipient's contact web site. The value of `<interactive>` indicates whether the URL is a web form known to require interaction.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

alert_id

Required. Specify the alert.

Type xsd:positiveInteger.

alert_type

Required. Specify the alert type.

Type advocacyAlertType.

Options are:

- action
- call
- lte

anonymous

Optional. Specify that the interaction is anonymous.

Type xsd:boolean.

Default is false.

body

Optional. Editable portion of message body. May be required by alert configuration.

Type xsd:string.

called

Optional. Call date. Required for call alerts. Ignored for other types of alerts.

Type xsd:date.

cc

Optional. Specify that the constituent wants to get a copy of the message sent to targets.

Type xsd:boolean.

Default is false.

city

Optional. Constituent city.

Type xsd:string.

contact_name

Optional. Call contact name. Optional for call alerts. Ignored for other types of alerts.

Type xsd:string.

contact_position

Optional. Call contact position. Required for call alerts. Ignored for other types of alerts.

Type xsd:string.

country

Optional. Constituent country.

Type xsd:string.

email

Optional. Constituent email address.

Type xsd:string.

first_name

Optional. Constituent first name.

Type xsd:string.

last_name

Optional. Constituent last name.

Type xsd:string.

note

Optional. Call note. Required for call alerts. Ignored for other types of alerts.

Type xsd:string.

opt_in

Optional. Specify that the constituent is opting in. The default is the same as the alert configuration.

Type xsd:boolean.

phone

Optional. Constituent phone. May be required by alert configuration.

Type xsd:string.

preview

Optional. Designates a test participation. Always specify preview=true when testing. When true, the participation will be processed and the result returned to the caller, but the messages will not be sent, and the interaction will not be logged in the database.

Type xsd:boolean.

Default is false.

recipient

Optional. Select recipients and delivery methods.

Type advocacyDelivery.

Options are:

- all
- internet
- fax
- print
- call

reply

Optional. Call reply. Required for call alerts. Ignored for other types of alerts.

Type xsd:string.

state

Optional. Constituent state.

Type xsd:string.

street1

Optional. Constituent street 1.

Type xsd:string.

street2

Optional. Constituent street 2.

Type xsd:string.

subject

Optional. Message subject. May be required by alert configuration.

Type xsd:string.

survey

Optional. Survey answers prefix.

Type xsd:string.

title

Optional. Constituent title. May be required by alert configuration.

Type xsd:string.

zip

Optional. Constituent zip.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Optional. Specifies the constituent associated with the alert or action.

Type xsd:nonNegativeInteger.

sign_redirects

Optional. Specifies that redirect URLs should be signed with a timestamp and a hash of the query string arguments and a secret key.

Type xsd:boolean.

Default is false.

alert_id

Required. Specify the alert.

Type xsd:positiveInteger.

alert_type

Required. Specify the alert type.

Type advocacyAlertType.

Options are:

- action
- call
- lte

anonymous

Optional. Specify that the interaction is anonymous.

Type xsd:boolean.

Default is false.

body

Optional. Editable portion of message body. May be required by alert configuration.

Type xsd:string.

called

Optional. Call date. Required for call alerts. Ignored for other types of alerts.

Type xsd:date.

cc

Optional. Specify that the constituent wants to get a copy of the message sent to targets.

Type xsd:boolean.

Default is false.

city

Optional. Constituent city.

Type xsd:string.

contact_name

Optional. Call contact name. Optional for call alerts. Ignored for other types of alerts.

Type xsd:string.

contact_position

Optional. Call contact position. Required for call alerts. Ignored for other types of alerts.

Type xsd:string.

country

Optional. Constituent country.

Type xsd:string.

email

Optional. Constituent email address.

Type xsd:string.

first_name

Optional. Constituent first name.

Type xsd:string.

last_name

Optional. Constituent last name.

Type xsd:string.

note

Optional. Call note. Required for call alerts. Ignored for other types of alerts.

Type xsd:string.

opt_in

Optional. Specify that the constituent is opting in. The default is the same as the alert configuration.

Type xsd:boolean.

phone

Optional. Constituent phone. May be required by alert configuration.

Type xsd:string.

preview

Optional. Designates a test participation. Always specify preview=true when testing. When true, the participation will be processed and the result returned to the caller, but the messages will not be sent, and the interaction will not be logged in the database.

Type xsd:boolean.

Default is false.

recipient

Optional. Select recipients and delivery methods.

Type advocacyDelivery.

Options are:

- all
- internet
- fax
- print
- call

reply

Optional. Call reply. Required for call alerts. Ignored for other types of alerts.

Type xsd:string.

state

Optional. Constituent state.

Type xsd:string.

street1

Optional. Constituent street 1.

Type xsd:string.

street2

Optional. Constituent street 2.

Type xsd:string.

subject

Optional. Message subject. May be required by alert configuration.

Type xsd:string.

survey

Optional. Survey answers prefix.

Type xsd:string.

title

Optional. Constituent title. May be required by alert configuration.

Type xsd:string.

zip

Optional. Constituent zip.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<takeActionResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <interaction>
    <interactionId>1001</interactionId>
    <interacted>2013-02-03T05:18:23.545-06:00</interacted>
    <alert>
      <alertId>103</alertId>
      <type>action</type>
      <status>active</status>
      <priority>high</priority>
      <url>https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&amp;id=103</url>
      <interactionCount>1000002</interactionCount>
      <title>Alert Title</title>
      <thumbnail>https://secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/103</thumbnail>
      <internalName>Internal Name</internalName>
      <description>Description</description>
      <category categoryId="1">General</category>
      <issues>
        <issue issueId="1001">Issue1</issue>
        <issue issueId="1002">Issue2</issue>
```

```
      </issues>
      <restrictByState>
        <state>TN</state>
        <state>TX</state>
      </restrictByState>
      <modified>2013-02-03T05:18:23.546-06:00</modified>
      <publish>2013-02-03T05:18:23.546-06:00</publish>
      <expire>2013-02-03T05:18:23.545-06:00</expire>
      <targets>
        <target>Governor</target>
        <target>US Senator</target>
        <target>US Representative</target>
        <target>State Senator</target>
        <target>State Representative</target>
      </targets>
      <messageSubject>Subject</messageSubject>
      <messageSubjectEditable>optional</messageSubjectEditable>
      <messageGreeting>Dear</messageGreeting>
      <messageOpening>Message Opening Text</messageOpening>
      <messageBody>Message Body Text</messageBody>
      <messageBodyEditable>optional</messageBodyEditable>
      <messageClosing>Sincerely,</messageClosing>
      <messageSignature>Signature</messageSignature>
      <wordLimit>0</wordLimit>
      <position>none</position>
      <yeaCount>0</yeaCount>
      <nayCount>0</nayCount>
      <abstainCount>0</abstainCount>
    </alert>
    <recipients>
      <recipient>
        <recipientId>ID</recipientId>
        <name>Senator John Q. Public</name>
        <url>http://public.senate.gov/contact.cfm</url>
        <interactive>true</interactive>
        <deliveryOptions>
          <delivery>internet</delivery>
        </deliveryOptions>
      </recipient>
    </recipients>
    <subject>Subject</subject>
    <body>Body</body>
    <called>2013-02-03T05:18:23.545-06:00</called>
    <contactPosition>ContactPosition</contactPosition>
    <contactName>ContactName</contactName>
    <reply>Reply</reply>
    <note>Note</note>
  </interaction>
</takeActionResponse>
```


JSON response


```
{"takeActionResponse":{"interaction":
{"body":"Body","interactionId":"1001","recipients":{"recipient":
{"recipientId":"ID","interactive":"true","name":"Senator John Q.
 Public","deliveryOptions":{"delivery":"internet"},"url":"http://
public.senate.gov/
contact.cfm"}},"contactName":"ContactName","subject":"Subject","alert":
{"position":"none","internalName":"Internal
 Name","messageSubjectEditable":"optional","messageSubject":"Subject","nayCount":"0","yea
 Title","alertId":"103","interactionCount":"1000002","description":"Description","priorit
```

```
{"target":["Governor","US Senator","US Representative","State Senator","State
 Representative"]},"messageGreeting":"Dear","messageBody":"Message Body
 Text","abstainCount":"0","expire":"2013-02-03T05:18:23.569-06:00","messageBodyEditable"
 Opening Text","url":"https://secure2.convio.net/orgname/site/Advocacy?
pagename=homepage&id=103","modified":"2013-02-03T05:18:23.570-06:00","messageClosing":"Si
{"content":"General","categoryId":"1"},"thumbnail":"https://
secure2.convio.net/orgname/images/Thumbnails/
ActionAlerts/103","restrictByState":{"state":
["TN","TX"]},"wordLimit":"0","issues":{"issue":
[{"content":"Issue1","issueId":"1001"},
{"content":"Issue2","issueId":"1002"}]}},"called":"2013-02-03T05:18:23.569-06:00","reply"
```


Examples

Survey Questions: DateQuestion

When returning a response to a `DateQuestion` type survey question, You must pass each part of the date value (the month, day, and year, and, optionally, the hour, minute, and second) as individual parameters to `takeAction`. For each date part, create separate parameter by appending one of the following to the the base survey parameter name: `_YEAR`, `_MONTH`, `_DAY`, `_HOUR24` (or `_HOUR12` and `_AMPM`), `_MINUTE`, `_SECOND`.

For example, given the following `DateQuestion` type survey question in the `getAdvocacyAlert` response:

```
<question>
<questionId>survey107_1000_1_1000</questionId>
<questionType>DateQuestion</questionType>
<text>Please update your DOB?</text>
<value/>
<required>true</required>
<questionChoices/>
<minSelected>0</minSelected>
<maxSelected>0</maxSelected>
<imageUrl/>
<audioUrl/>
</question>
```

You might render the survey question in an HTML form as:

```
<label for="survey107_1000_1_1000">Please update your DOB?</label><br />
  <fieldgroup id="survey107_1000_1_1000">
  <input type="text" name="survey107_1000_1_1000_MONTH" size="2" value="MM" />
 -
  <input type="text" name="survey107_1000_1_1000_DAY" size="2" value="DD" /> -
  <input type="text" name="survey107_1000_1_1000_YEAR" size="4" value="YYYY" /
> <br />
</fieldgroup>
```

The value posted to `takeAction` representing the response would look like:

```
survey107_1000_1_1000_YEAR=2010&survey107_1000_1_1000_MONTH=6&survey107_1000_1_1000_DAY=1
```

Survey Questions: ComboChoice

A `ComboChoice` survey question presents users two options. They can either pick from a list of predefined values, or they can supply an alternate value not on the list. This is rendered using an HTML "select" type input for the predefined list, a "text" input for the alternate value, and "radio" type inputs to choose between them.

You must return each part of a `ComboChoice` survey question as an individual parameter by postfixing the word "survey" in the the base survey parameter name with `radio_`, `select_`, or `text_`, as appropriate.

For example, given the following `ComboChoice` question in the `getAdvocacyAlert` response:

```
<question>
<questionId>survey107_1000_2_1001</questionId>
<questionType>ComboChoice</questionType>
<text>Please update your suffix?</text>
<value/>
<required>false</required>
<questionChoices>
<choice name="Sr." selected="false">Sr.</choice>
<choice name="Jr." selected="false">Jr.</choice>
<choice name="II" selected="false">II</choice>
<choice name="III" selected="false">III</choice>
<choice name="IV" selected="false">IV</choice>
</questionChoices>
<minSelected>0</minSelected>
<maxSelected>0</maxSelected>
<imageUrl/>
<audioUrl/>
</question>
```

You might render the survey question in an HTML form as:

```
<label for="survey107_1000_2_1001">Please update your suffix?</label><br />
<fieldgroup id="survey107_1000_2_1001">
  <input type="radio" name="surveyradio_107_1000_2_1001" value="1">
  <select name="surveyselect_107_1000_2_1001">
    <option>Sr.<option>Jr.<option>II<option>III<option>IV
  </select>
  <input type="radio" name="surveyradio_107_1000_2_1001" value="2">
  <input type="text" name="surveytext_107_1000_2_1001">
</fieldgroup>
```

If the user selects one of the predefined values, the parameters posted to `takeAction` should look like this:

```
surveyselect_107_1000_2_1001=Sr.&surveyradio_107_1000_2_1001=1
```

If the user supplies a different value, the parameters posted to `takeAction` should look like this:

```
surveytext_107_1000_2_1001=PhD.&surveyradio_107_1000_2_1001=2
```

If you specify only a `surveytext` value, that value will be used whether it is one of the predefined choices or another value:

```
surveytext_107_1000_2_1001=PhD.
```

Constituent API

CRConsAPI
`https://secure2.convio.net/organization/site/CRConsAPI`
SRConsAPI
`https://secure2.convio.net/organization/site/SRConsAPI`
Description

The Constituent Management API communicates directly with the Convio Constituent360 database and provides methods for all of the following:

- Create New Constituent Records - create new records in the Constituent360 database
- Fetch Constituent Records - request XML or JSON formatted biographical data for a constituent record stored in the Constituent360 database
- Update Constituent Records - update constituent records in the Constituent360 database
- Manage Group Membership and Subscriptions - manage constituent group memberships (fetch/add/remove) and interest category preferences (fetch/add/remove)
- Log User Interactions - log custom interactions in response to activities performed by a user within your application

API Administrator accounts used for Server API calls to this API must belong to an Administrative Security Group that has "Use Convio APIs" permission in the "Contact Management" permission type. The built-in "API Administrators" group has this permission by default.

The Constituent API exposes the following methods:

create

This method creates a new user record and returns the record's cons_id unique identifier.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Server.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=create &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&add_center_ids=`*value*] [`&add_center_opt_in_ids=`*value*] [`&add_group_ids=`*value*] [`&add_interest_ids=`*value*] [`&get_single_sign_on_token=`*value*] [`&interaction_body=`*value*] [`&interaction_cat_id=`*value*] [`&interaction_count=`*value*] [`&interaction_subject=`*value*] [`&member_id=`*value*] [`&no_welcome=`*value*] [`&primary_email=`*value*] [`&suppress_cleaning=`*value*] [`&teamraiser_registration=`*value*]
Server API Syntax
`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=create & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] [`& add_center_ids =`*value*] [`& add_center_opt_in_ids =`*value*] [`& add_group_ids =`*value*] [`& add_interest_ids =`*value*] [`& get_single_sign_on_token =`*value*] [`& interaction_body =`*value*] [`& interaction_cat_id =`*value*] [`& interaction_count =`*value*] [`& interaction_subject =`*value*] [`& member_id =`*value*] [`& no_welcome =`*value*] [`& primary_email =`*value*] [`& suppress_cleaning =`*value*] [`& teamraiser_registration =`*value*]
Usage Notes

The `create` method adds a new constituent record to the Convio Constituent 360 database with the following initial state:

- origin = "Constituent API" (35)

- active = "Active" (1)
- active_detail = "Known" (2) if a "welcome" auto-responder is sent, "Unknown" (1) if not sent.
- donor_status = "Non-Donor" (1)

The Client API version of this method does not require authentication of the caller. The list of constituent record fields that can be accessed or set is customizable for each organization, and is different for administrative users and non-administrative users.

listUserFields will return the list of the Constituent record fields the caller may specify as parameters on `create`.

Sending a "welcome" auto-responder by specifying the parameter 'no_welcome=false' also sets 'active_detail=known' for the constituent's initial state.

This method automatically invokes *logInteraction* to log an interaction for the target constituent.

Note: If successful, the client API establishes a session for the user and pushes a session cookie to the caller. A non-administrative user in an active session is not permitted to create other user records; subsequent calls to `create` will fail, returning an authorization error. If you use the client API to create more than one user at a time, you may have to call *logout* between each `create` to clear the last-created user's session.

Note: This method will not allow you to add the created constituent directly to an Interest Opt-In or Opt-Out group. You should instead add or remove the interest to the constituent, which will adjust the group membership as appropriate.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

add_center_ids

Optional. Comma-separated list of numeric IDs of centers to add the user to.

Type xsd:string.

add_center_opt_in_ids

Optional. Comma-separated list of numeric IDs of centers for which the user should be opted in to email communications.

Type xsd:string.

add_group_ids

Optional. Comma-separated list of numeric IDs of groups to add the user to. Only available when called using an administrator account. Cannot be used for pre-defined groups (those with IDs less than 1000). Cannot be used for administrator security groups.

Type xsd:string.

add_interest_ids

Optional. Comma-separated list of numeric IDs of email interests to add the user to.

Type xsd:string.

get_single_sign_on_token

Optional. Specifies whether a single-sign-on token should be returned with the response to the create method.

Type xsd:boolean.

interaction_body

Optional. Text describing or associated with the interaction.

Type xsd:string.

interaction_cat_id

Optional. The Security Category associated with this interaction. The default is 0, which corresponds to the "General" category.

Type xsd:nonNegativeInteger.

Default is 0.

interaction_count

Optional. The number of times the constituent performed this interaction.

Type xsd:nonNegativeInteger.

Default is 1.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

no_welcome

Optional. Only used when creating new records. Specifies whether to send a "welcome" auto-responder email to the new user. If the value is "f" or "false" (ignoring case) and the site is configured to send welcome auto-responders to new users, then one will be sent. By default, this value is "true", meaning that no auto-responder email will be sent.

Type xsd:boolean.

Default is true.

primary_email

Optional. The primary email address that uniquely identifies the constituent. Required to create a new constituent record. Optional when updating a record.

Type xsd:string.

suppress_cleaning

Optional. Specifies whether data cleaning/standardization should be turned on or off. This parameter is intended for performance tuning for server side batch APIs calls.

Type xsd:boolean.

teamraiser_registration

Optional. Specifies that this is a team raiser registration and should process the consituent create with Team Raiser site settings. The default value is false.

Type xsd:boolean.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

add_center_ids

Optional. Comma-separated list of numeric IDs of centers to add the user to.

Type xsd:string.

add_center_opt_in_ids

Optional. Comma-separated list of numeric IDs of centers for which the user should be opted in to email communications.

Type xsd:string.

add_group_ids

Optional. Comma-separated list of numeric IDs of groups to add the user to. Only available when called using an administrator account. Cannot be used for pre-defined groups (those with IDs less than 1000). Cannot be used for administrator security groups.

Type xsd:string.

add_interest_ids

Optional. Comma-separated list of numeric IDs of email interests to add the user to.

Type xsd:string.

get_single_sign_on_token

Optional. Specifies whether a single-sign-on token should be returned with the response to the create method.

Type xsd:boolean.

interaction_body

Optional. Text describing or associated with the interaction.

Type xsd:string.

interaction_cat_id

Optional. The Security Category associated with this interaction. The default is 0, which corresponds to the "General" category.

Type xsd:nonNegativeInteger.

Default is 0.

interaction_count

Optional. The number of times the constituent performed this interaction.

Type xsd:nonNegativeInteger.

Default is 1.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

no_welcome

Optional. Only used when creating new records. Specifies whether to send a "welcome" auto-responder email to the new user. If the value is "f" or "false" (ignoring case) and the site is

configured to send welcome auto-responders to new users, then one will be sent. By default, this value is "true", meaning that no auto-responder email will be sent.

Type xsd:boolean.

Default is true.

primary_email

Optional. The primary email address that uniquely identifies the constituent. Required to create a new constituent record. Optional when updating a record.

Type xsd:string.

suppress_cleaning

Optional. Specifies whether data cleaning/standardization should be turned on or off. This parameter is intended for performance tuning for server side batch APIs calls.

Type xsd:boolean.

teamraiser_registration

Optional. Specifies that this is a team raiser registration and should process the consituent create with Team Raiser site settings. The default value is false.

Type xsd:boolean.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<createConsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cons_id>1001002</cons_id>
  <message>User created.</message>
</createConsResponse>
```

JSON response

```
{"createConsResponse":{"message":"User created.","cons_id":"1001002"}}
```

createOrUpdate

This method provides a convenience wrapper around the create and update APIs. When called, this method will first try to locate and update an existing record, and if no existing record can be found, will create a new record.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=createOrUpdate`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*

][&add_center_ids=*value*][&add_center_opt_in_ids=*value*][&add_group_ids=*value*][&add_interest_ids=*value*][&cons_id=*value*][&interaction_body=*value*][&interaction_cat_id=*value*][&interaction_count=*value*][&interaction_subject=*value*][&member_id=*value*][&no_welcome=*value*][&primary_email=*value*][&remove_center_ids=*value*][&remove_center_opt_in_ids=*value*][&remove_group_ids=*value*][&remove_interest_ids=*value*][&suppress_cleaning=*value*][&auth=*value*][&sso_auth_token=*value*]

Server API Syntax

https://*secure2.convio.net/organization*/site/SRConsAPI?method=createOrUpdate & api_key =*value* & login_name =*value* & login_password =*value* & v =*value* [& center_id =*value*] [& response_format =*xml* |*json*] [& source =*value*] [& sub_source =*value*] [& suppress_response_codes =*value*] [& add_center_ids =*value*] [& add_center_opt_in_ids =*value*] [& add_group_ids =*value*] [& add_interest_ids =*value*] [& cons_id =*value*] [& interaction_body =*value*] [& interaction_cat_id =*value*] [& interaction_count =*value*] [& interaction_subject =*value*] [& member_id =*value*] [& no_welcome =*value*] [& primary_email =*value*] [& remove_center_ids =*value*] [& remove_center_opt_in_ids =*value*] [& remove_group_ids =*value*] [& remove_interest_ids =*value*] [& suppress_cleaning =*value*]

Usage Notes

The `createOrUpdate` method provides a convenience wrapper around *create* and *update*. This method attempts to locate an existing constituent record in the Convio Constituent 360 database using the value of the cons_id, member_id, or primary_email parameters, and either updates the record, if found, or creates a new record with if possible with the information provided.

If called as a Client API, this method requires authentication. If the Client API operates on behalf of a logged-in non-administrative user (regular constituent), it has the same effect as the `update` method operating on the logged-in user's record. The Client API for this method cannot be used in the context of a non-administrative user to create a record; use the `create` method instead.

If a Server API call or Client API call using the authorization token of an administrator session fails to locate an existing record, it will attempt to create a new record for the constituent with the following initial state:

- origin = "Constituent API" (35)
- active = "Active" (1)
- active_detail = "Known" (2) if a "welcome" auto-responder is sent, "Unknown" (1) if not sent.
- donor_status = "Non-Donor" (1)

Notes:

1. If a cons_id is included and a match is found, the record is updated. If no match is found, an error is returned indicating that the specified record does not exist.
2. If the record requested by member_id or primary_email and is not found, the system will attempt to add a new record as by the `create` method.
3. If a match is found for member_id, and there is only one match, then the record is updated; if more than one record is matched, an error is returned.
4. If a match is found for primary_email, and there is only one match, the record is updated; if more than one record is matched, and: If the site is configured to enforce unique email, then an error is returned. If enforce unique email is not enabled, then a new record is created.

The list of constituent record fields that can be set is customizable for each organization. It is different for administrative users (via the Server API) and non-administrative users (via Client API).

The *listUserFields* method will return the list of the Constituent record fields the caller may view or add.

When creating a new record by this method, sending a "welcome" auto-responder by specifying the parameter 'no_welcome=false' also sets 'active_detail=known' for the constituent's initial state.

This method automatically invokes *logInteraction* to log an interaction for the target constituent.

Note: This method will not allow you to add the created constituent directly to an Interest Opt-In or Opt-Out group.

You should instead add or remove the interest to the constituent, which will adjust the group membership as appropriate.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

add_center_ids

Optional. Comma-separated list of numeric IDs of centers to add the user to.

Type xsd:string.

add_center_opt_in_ids

Optional. Comma-separated list of numeric IDs of centers for which the user should be opted in to email communications.

Type xsd:string.

add_group_ids

Optional. Comma-separated list of numeric IDs of groups to add the user to. Only available when called using an administrator account. Cannot be used for pre-defined groups (those with IDs less than 1000). Cannot be used for administrator security groups.

Type xsd:string.

add_interest_ids

Optional. Comma-separated list of numeric IDs of email interests to add the user to.

Type xsd:string.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

interaction_body

Optional. Text describing or associated with the interaction.

Type xsd:string.

interaction_cat_id

Optional. The Security Category associated with this interaction. The default is 0, which corresponds to the "General" category.

Type xsd:nonNegativeInteger.

Default is 0.

interaction_count

Optional. The number of times the constituent performed this interaction.

Type xsd:nonNegativeInteger.

Default is 1.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

no_welcome

Optional. Only used when creating new records. Specifies whether to send a "welcome" auto-responder email to the new user. If the value is "f" or "false" (ignoring case) and the site is configured to send welcome auto-responders to new users, then one will be sent. By default, this value is "true", meaning that no auto-responder email will be sent.

Type xsd:boolean.

Default is true.

primary_email

Optional. The primary email address that uniquely identifies the constituent. Required to create a new constituent record. Optional when updating a record.

Type xsd:string.

remove_center_ids

Optional. Comma-separated list of numeric IDs of centers to remove the user from.

Type xsd:string.

remove_center_opt_in_ids

Optional. Comma-separated list of numeric IDs of centers for which the user should be opted out of email communications.

Type xsd:string.

remove_group_ids

Optional. Comma-separated list of numeric IDs of groups to remove the user from. Only valid when updating records. Only available when called using an administrator account. Cannot be used for pre-defined groups (those with IDs less than 1000). Cannot be used for administrator security groups.

Type xsd:string.

remove_interest_ids

Optional. Comma-separated list of numeric IDs of email interests to remove the user from. Only valid when updating records.

Type xsd:string.

suppress_cleaning

Optional. Specifies whether data cleaning/standardization should be turned on or off. This parameter is intended for performance tuning for server side batch APIs calls.

Type xsd:boolean.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

add_center_ids

Optional. Comma-separated list of numeric IDs of centers to add the user to.

Type xsd:string.

add_center_opt_in_ids

Optional. Comma-separated list of numeric IDs of centers for which the user should be opted in to email communications.

Type xsd:string.

add_group_ids

Optional. Comma-separated list of numeric IDs of groups to add the user to. Only available when called using an administrator account. Cannot be used for pre-defined groups (those with IDs less than 1000). Cannot be used for administrator security groups.

Type xsd:string.

add_interest_ids

Optional. Comma-separated list of numeric IDs of email interests to add the user to.

Type xsd:string.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

interaction_body

Optional. Text describing or associated with the interaction.

Type xsd:string.

interaction_cat_id

Optional. The Security Category associated with this interaction. The default is 0, which corresponds to the "General" category.

Type xsd:nonNegativeInteger.

Default is 0.

interaction_count

Optional. The number of times the constituent performed this interaction.

Type xsd:nonNegativeInteger.

Default is 1.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

no_welcome

Optional. Only used when creating new records. Specifies whether to send a "welcome" auto-responder email to the new user. If the value is "f" or "false" (ignoring case) and the site is configured to send welcome auto-responders to new users, then one will be sent. By default, this value is "true", meaning that no auto-responder email will be sent.

Type xsd:boolean.

Default is true.

primary_email

Optional. The primary email address that uniquely identifies the constituent. Required to create a new constituent record. Optional when updating a record.

Type xsd:string.

remove_center_ids

Optional. Comma-separated list of numeric IDs of centers to remove the user from.

Type xsd:string.

remove_center_opt_in_ids

Optional. Comma-separated list of numeric IDs of centers for which the user should be opted out of email communications.

Type xsd:string.

remove_group_ids

Optional. Comma-separated list of numeric IDs of groups to remove the user from. Only valid when updating records. Only available when called using an administrator account. Cannot be used for pre-defined groups (those with IDs less than 1000). Cannot be used for administrator security groups.

Type xsd:string.

remove_interest_ids

Optional. Comma-separated list of numeric IDs of email interests to remove the user from. Only valid when updating records.

Type xsd:string.

suppress_cleaning

Optional. Specifies whether data cleaning/standardization should be turned on or off. This parameter is intended for performance tuning for server side batch APIs calls.

Type xsd:boolean.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<createOrUpdateConsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cons_id>1001002</cons_id>
  <message>User updated.</message>
</createOrUpdateConsResponse>
```

JSON response

```
{"createOrUpdateConsResponse":{"message":"User updated.","cons_id":"1001002"}}
```

getGroupMembers

This method returns the list of the constituents in a group. By default, each group member's constituent ID, member ID, first and last names, primary email address and home address are returned.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=getGroupMembers &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&group_id=`*value* [`&fields=`*value*] [`&max_number=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=getGroupMembers & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& group_id =`*value* [`& fields =`*value*] [`& max_number =`*value*]

Usage Notes

This method returns personal information for members of a user group. It is only available for administrative users having the appropriate permission to review member lists. If this method is called as a Client API, the administrator must be logged in and an authentication token must be provided in the request.

By default, the method returns constituent ID, member ID, name, primary mailing address and primary email address. The fields parameter can be used to specify a comma-separated list of other fields from the constituent record.The *listUserFields* method returns the list of available fields, which may differ from one Convio site to another.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

group_id

Required. The Group ID key referencing the Group record.

Type xsd:nonNegativeInteger.

fields

Optional. A comma-separated list of field names of constituent record fields to return for each member.

Type xsd:string.

Default is cons_id, member_id, first_name, last_name, primary_email, home_street1, home_street2, home_street3, home_city, home_stateprov, home_county, home_zip.

max_number

Optional. The maximum number of members to return (up to 1000).

Type xsd:nonNegativeInteger.

Default is 1000.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

group_id

Required. The Group ID key referencing the Group record.

Type xsd:nonNegativeInteger.

fields

Optional. A comma-separated list of field names of constituent record fields to return for each member.

Type xsd:string.

Default is cons_id, member_id, first_name, last_name, primary_email, home_street1, home_street2, home_street3, home_city, home_stateprov, home_county, home_zip.

max_number

Optional. The maximum number of members to return (up to 1000).

Type xsd:nonNegativeInteger.

Default is 1000.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getGroupMembersResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <total_number>2</total_number>
  <number_returned>2</number_returned>
  <member>
    <cons_id>1000121</cons_id>
    <member_id>Member ID</member_id>
    <name>
      <first>First</first>
      <last>Last</last>
    </name>
    <email>
      <primary_address>email@address.net</primary_address>
    </email>
    <primary_address>
      <street1>Street1</street1>
      <street2>Street2</street2>
      <street3>Street3</street3>
      <city>City</city>
      <state>State</state>
      <zip>Zip</zip>
      <county>County</county>
    </primary_address>
  </member>
</getGroupMembersResponse>
```

JSON response

```
{"getGroupMembersResponse":{"member":{"primary_address":
{"street2":"Street2","zip":"Zip","street1":"Street1","county":"County","state":"State","s
{"primary_address":"email@address.net"},"name":
```

```
{"last":"Last","first":"First"},"cons_id":"1000121","member_id":"Member
 ID"},"number_returned":"2","total_number":"2"}}
```

getInteraction

This method can be used to get an interaction for a constituent. It must be called by a logged-in user within an authenticated session. This method requires at least one unique identifier for the user (may be cons_id, member_id, or primary_email).

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Server.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=getInteraction`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&interaction_id=`*value* [`&cons_id=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*]
Server API Syntax
`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=getInteraction &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& interaction_id =`*value* [`& cons_id =`*value*] [`& member_id =`*value*] [`& primary_email =`*value*]

Usage Notes

If called as a Client API, this method requires authentication. If the requester is an administrator (normally only the case using the Server API):

1. One of cons_id, member_id, or primary_email must be specified in order to locate the record.
2. The order of precedence of the search terms is (a) cons_id (b) member_id, then (c) primary_email. If a record is not found or if multiple records are returned by any specified search term evaluated in this order, the method returns an error.

If the requester is a constituent using the Client API:

1. An authentication token representing the logged-in user must be passed.
2. Only the constituent record of the logged in user may be retrieved.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

interaction_id

Required. The id of the interaction.

Type xsd:nonNegativeInteger.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

interaction_id

Required. The id of the interaction.

Type xsd:nonNegativeInteger.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getInteractionResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <interaction>
    <interactionId>1000</interactionId>
    <interactionTypeId>1</interactionTypeId>
    <categoryId>0</categoryId>
    <referenceId>0</referenceId>
    <consId>1001001</consId>
    <createdBy>1001001</createdBy>
    <creationDate>2013-02-03T05:18:20.343-06:00</creationDate>
    <note>
      <categoryId>0</categoryId>
      <typeOfConsNote>normal</typeOfConsNote>
      <consId>1001001</consId>
      <createdBy>1001001</createdBy>
      <creationDate>2013-02-03T05:18:20.343-06:00</creationDate>
      <interactionId>1000</interactionId>
      <modifiedBy>1001001</modifiedBy>
      <modifiedDate>2013-02-03T05:18:20.343-06:00</modifiedDate>
      <noteId>1000</noteId>
      <subject>Sample Subject</subject>
```

```
      <text>Lorem ipsum dolor sit amet, consetetur sadipscing</text>
      <type>comment</type>
    </note>
    <subject>Sample Subject</subject>
    <count>1</count>
  </interaction>
</getInteractionResponse>
```

JSON response

```
{"getInteractionResponse":{"interaction":
{"creationDate":"2013-02-03T05:18:20.343-06:00","interactionTypeId":"1","count":"1","crea
 Subject","categoryId":"0","referenceId":"0","note":
{"creationDate":"2013-02-03T05:18:20.343-06:00","text":"Lorem ipsum dolor sit
 amet, consetetur
 sadipscing","interactionId":"1000","createdBy":"1001001","consId":"1001001","typeOfConsN
 Subject","modifiedBy":"1001001","categoryId":"0","noteId":"1000","type":"comment","modif
```

getUser

This method returns a single constituent record.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=getUser &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&cons_id=`*value*] [`&fields=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=getUser & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] [`& cons_id =`*value*] [`& fields =`*value*] [`& member_id =`*value*] [`& primary_email =`*value*]

Usage Notes

If called as a Client API, this method requires authentication. If the requester is an administrator (normally only the case using the Server API):

1. One of cons_id, member_id, or primary_email must be specified in order to locate the record.
2. The order of precedence of the search terms is (a) cons_id (b) member_id, then (c) primary_email. If a record is not found or if multiple records are returned by any specified search term evaluated in this order, the method returns an error.

If the requester is a constituent using the Client API:

1. An authentication token representing the logged-in user must be passed.
2. Only the constituent record of the logged in user may be retrieved.

If a field is not included in the response, it is either not enabled for the site or it is invisible to the caller. If the field is included but has no value (xsi:nil="true" in an XML response or null in a JSON response), then the field is visible

and has a null value for the retrieved record. If a composite element (one that can contain nested elements), such as "name" or "email" in the example above, is nil then all of its contained fields are either disabled or have nil values for this record.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

fields

Optional. Specifies a comma-separated list of names of fields to retrieve. If not specified, then all fields in use and visible to the caller are returned. Use "dot" notation to refer to nested elements of the XML schema. For example, use "name.first" to request the "first" element of the "name" element (the user's first name).

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

fields

Optional. Specifies a comma-separated list of names of fields to retrieve. If not specified, then all fields in use and visible to the caller are returned. Use "dot" notation to refer to nested elements of the XML schema. For example, use "name.first" to request the "first" element of the "name" element (the user's first name).

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getConsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cons_id>1000121</cons_id>
  <member_id>Member ID</member_id>
  <user_name>Username</user_name>
  <name>
    <first>First</first>
    <last>Last</last>
  </name>
  <email>
    <primary_address>email@address.net</primary_address>
  </email>
  <primary_address>
    <street1>Street</street1>
    <city>City</city>
    <state>State</state>
    <zip>Zip</zip>
  </primary_address>
</getConsResponse>
```

JSON response

```
{"getConsResponse":{"user_name":"Username","primary_address":
{"zip":"Zip","street1":"Street","state":"State","city":"City"},"email":
{"primary_address":"email@address.net"},"name":
{"last":"Last","first":"First"},"cons_id":"1000121","member_id":"Member ID"}}
```

getUserCenters

This method returns a list of centers that the specified constituent is a member of.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=getUserCenters`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&cons_id=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=getUserCenters &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`&`

`suppress_response_codes` =*value*] [& `cons_id` =*value*] [& `member_id` =*value*] [& `primary_email` =*value*]

Usage Notes

If this method is called within a logged-in session context, either from a "remember me" cookie or a user login, this method will return a list of the Centers the constituent belongs to, if any.

If the caller is not within a logged-in session context, this method requires a cons_id, a member_id, or a primary_email parameter that uniquely identifies the constituent, and returns the list of Centers associated with that constituent.

If no constituent or if multiple constituents match the information supplied, or if the Convio site is not configured for Multi-Center support, this method retuns an empty result set.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getConsCentersResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <centerMembership>
    <center>
      <id>1</id>
      <label>Center Name</label>
      <description>Center Description</description>
      <home_page>http://www.your_org.org/site/PageServer?
pagename=center_home</home_page>
      <not_found_page>http://www.your_org.org/site/PageServer?
pagename=center_not_found</not_found_page>
    </center>
    <email_opt_in>true</email_opt_in>
  </centerMembership>
</getConsCentersResponse>
```

JSON response

```
{"getConsCentersResponse":{"centerMembership":{"center":
{"id":"1","description":"Center Description","label":"Center
 Name","home_page":"http://www.your_org.org/site/PageServer?
pagename=center_home","not_found_page":"http://www.your_org.org/site/
PageServer?pagename=center_not_found"},"email_opt_in":"true"}}}
```

getUserGroups

This method returns a list of the IDs and names of groups that the specified constituent belongs to.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=getUserGroups &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&cons_id=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=getUserGroups &api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] [`&cons_id=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*]

Usage Notes

If called as a Client API, this method requires authentication. If the requester is an administrator (normally only the case using the Server API):

1. One of cons_id, member_id, or primary_email must be specified in order to locate the record.

2. The order of precedence of the search terms is (a) cons_id (b) member_id, then (c) primary_email. If a record is not found or if multiple records are returned by any specified search term evaluated in this order, the method returns an error.

If the requester is a constituent using the Client API:

1. An authentication token representing the logged-in user must be passed.
2. Only groups associated with the logged in user may be retrieved.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getConsGroupsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
  <group>
    <id>1001</id>
    <label>Group Name</label>
  </group>
</getConsGroupsResponse>
```

JSON response

```
{"getConsGroupsResponse":{"group":{"id":"1001","label":"Group Name"}}}
```

getUserInteractions

This method can be used to get interactions for a constituent. It must be called by a logged-in user within an authenticated session. This method requires at least one unique identifier for the user (may be cons_id, member_id, or primary_email). Only notes for custom interactions are returned. Notes for standard interactions can be retrieved using the getInteraction API method.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Server.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=getUserInteractions`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&begin_date=`*value*] [`&cons_id=`*value*] [`&end_date=`*value*] [`&interaction_subject=`*value*] [`&interaction_type_id=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*]
Server API Syntax
`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=getUserInteractions`
`&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] [`&begin_date=`*value*] [`&cons_id=`*value*] [`&end_date=`*value*] [`&interaction_subject=`*value*] [`&interaction_type_id=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*]

Usage Notes

If called as a Client API, this method requires authentication. If the requester is an administrator (normally only the case using the Server API):

1. One of cons_id, member_id, or primary_email must be specified in order to locate the record.
2. The order of precedence of the search terms is (a) cons_id (b) member_id, then (c) primary_email. If a record is not found or if multiple records are returned by any specified search term evaluated in this order, the method returns an error.

If the requester is a constituent using the Client API:

1. An authentication token representing the logged-in user must be passed.

2. Only the constituent record of the logged in user may be retrieved.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

begin_date

Optional. The beginning of the date range of the interactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

end_date

Optional. The ending of the date range of the interactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

interaction_type_id

Optional. The Interaction Type associated with this interaction.

Type xsd:nonNegativeInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

begin_date

Optional. The beginning of the date range of the interactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

end_date

Optional. The ending of the date range of the interactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

interaction_type_id

Optional. The Interaction Type associated with this interaction.

Type xsd:nonNegativeInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<getUserInteractionsResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <interaction>
    <interactionId>1000</interactionId>
    <interactionTypeId>1000</interactionTypeId>
    <categoryId>1000</categoryId>
    <referenceId>0</referenceId>
    <consId>1001001</consId>
    <createdBy>1001001</createdBy>
    <creationDate>2013-02-03T05:18:20.605-06:00</creationDate>
    <subject>Sample Subject 0</subject>
    <count>1</count>
    <note>
      <categoryId>1000</categoryId>
      <typeOfConsNote>normal</typeOfConsNote>
      <consId>1001001</consId>
      <createdBy>1001001</createdBy>
      <creationDate>2013-02-03T05:18:20.605-06:00</creationDate>
      <interactionId>1000</interactionId>
      <modifiedBy>1001001</modifiedBy>
      <modifiedDate>2013-02-03T05:18:20.605-06:00</modifiedDate>
      <noteId>1000</noteId>
      <subject>Sample Subject 0</subject>
      <text>Lorem ipsum dolor sit amet, consetetur sadipscing 0</text>
      <type>comment</type>
    </note>
  </interaction>
  <interaction>
    <interactionId>1001</interactionId>
    <interactionTypeId>1001</interactionTypeId>
    <categoryId>1001</categoryId>
    <referenceId>0</referenceId>
    <consId>1001002</consId>
    <createdBy>1001002</createdBy>
    <creationDate>2013-02-03T05:18:20.605-06:00</creationDate>
    <subject>Sample Subject 1</subject>
    <count>2</count>
    <note>
      <categoryId>1001</categoryId>
      <typeOfConsNote>normal</typeOfConsNote>
      <consId>1001002</consId>
      <createdBy>1001002</createdBy>
      <creationDate>2013-02-03T05:18:20.605-06:00</creationDate>
      <interactionId>1001</interactionId>
      <modifiedBy>1001002</modifiedBy>
      <modifiedDate>2013-02-03T05:18:20.605-06:00</modifiedDate>
      <noteId>1001</noteId>
      <subject>Sample Subject 1</subject>
      <text>Lorem ipsum dolor sit amet, consetetur sadipscing 1</text>
      <type>comment</type>
    </note>
  </interaction>
  <interaction>
    <interactionId>1002</interactionId>
    <interactionTypeId>6</interactionTypeId>
    <categoryId>1002</categoryId>
    <referenceId>0</referenceId>
    <consId>1001003</consId>
    <createdBy>1001003</createdBy>
    <creationDate>2013-02-03T05:18:20.605-06:00</creationDate>
    <subject>Sample Subject 2</subject>
    <count>3</count>
  </interaction>
```

```
</getUserInteractionsResponse>
```

JSON response


```
{"getUserInteractionsResponse":{"interaction":
[{"creationDate":"2013-02-03T05:18:20.605-06:00","interactionTypeId":"1000","count":"1","
 Subject 0","categoryId":"1000","referenceId":"0","note":
{"creationDate":"2013-02-03T05:18:20.605-06:00","text":"Lorem ipsum dolor sit
 amet, consetetur sadipscing
 0","interactionId":"1000","createdBy":"1001001","consId":"1001001","typeOfConsNote":"no
 Subject
 0","modifiedBy":"1001001","categoryId":"1000","noteId":"1000","type":"comment","modifie
{"creationDate":"2013-02-03T05:18:20.605-06:00","interactionTypeId":"1001","count":"2","
 Subject 1","categoryId":"1001","referenceId":"0","note":
{"creationDate":"2013-02-03T05:18:20.605-06:00","text":"Lorem ipsum dolor sit
 amet, consetetur sadipscing
 1","interactionId":"1001","createdBy":"1001002","consId":"1001002","typeOfConsNote":"no
 Subject
 1","modifiedBy":"1001002","categoryId":"1001","noteId":"1001","type":"comment","modifie
{"creationDate":"2013-02-03T05:18:20.605-06:00","interactionTypeId":"6","count":"3","crea
 Subject 2","categoryId":"1002","referenceId":"0"}]}}
```


getUserInterests

This method returns a list of interest categories associated with the specified constituent.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=getUserInterests`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&cons_id=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=getUserInterests &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] [`& cons_id =`*value*] [`& member_id =`*value*] [`& primary_email =`*value*]

Usage Notes

If called as a Client API, this method requires authentication. If the requester is an administrator (normally only the case using the Server API):

1. One of cons_id, member_id, or primary_email must be specified in order to locate the record.
2. The order of precedence of the search terms is (a) cons_id (b) member_id, then (c) primary_email. If a record is not found or if multiple records are returned by any specified search term evaluated in this order, the method returns an error.

If the requester is a constituent using the Client API:

1. An authentication token representing the logged-in user must be passed.
2. Only the constituent record of the logged in user may be retrieved.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getConsInterestsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <interest>
    <id>1</id>
    <label>Interest Name</label>
    <path>Interest Full Name</path>
    <for_email>false</for_email>
    <for_web>true</for_web>
    <description>Interest Description</description>
  </interest>
</getConsInterestsResponse>
```

JSON response

```
{"getConsInterestsResponse":{"interest":{"id":"1","description":"Interest
 Description","for_web":"true","path":"Interest Full
 Name","for_email":"false","label":"Interest Name"}}}
```

getUserTransactions

This method returns transactions for a single constituent record, optionally filtered by transaction type and date range.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=getUserTransactions`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&begin_date=`*value*] [`&cons_id=`*value*] [`&end_date=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*] [`&transaction_type=`*DONATION* | *PLEDGE* | *RECURRING_GIFT* | *TICKET* | *DONATION_PENDING* | *DONATION_PENDING_ACCEPTED* | *DONATION_PENDING_REJECTED* | *DONATION_PENDING_ACCEPTED_FAILED* | *DONATION_PENDING_REJECTED_FAILED* | *DONATION_ACCEPTED_FAILED* | *DONATION_REJECTED_FAILED* | *DONATION_SETTLED* | *DONATION_REJECTED* | *F2F_REGISTRATION* | *TRIBUTES_REGISTRATION* | *ECOMMERCE_PURCHASE* | *REFUND* | *FAILED_RECURRING_DONATION_DECLINED* | *FAILED_RECURRING_DONATION_FRAUD*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=getUserTransactions`
`&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] [`&begin_date=`*value*] [`&cons_id=`*value*] [`&end_date=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*] [`&transaction_type=`*DONATION* | *PLEDGE* | *RECURRING_GIFT* | *TICKET* | *DONATION_PENDING* | *DONATION_PENDING_ACCEPTED* | *DONATION_PENDING_REJECTED* | *DONATION_PENDING_ACCEPTED_FAILED* | *DONATION_PENDING_REJECTED_FAILED* | *DONATION_ACCEPTED_FAILED* | *DONATION_REJECTED_FAILED* | *DONATION_SETTLED* | *DONATION_REJECTED* | *F2F_REGISTRATION* | *TRIBUTES_REGISTRATION* | *ECOMMERCE_PURCHASE* | *REFUND* | *FAILED_RECURRING_DONATION_DECLINED* | *FAILED_RECURRING_DONATION_FRAUD*]

Usage Notes

If called as a Client API, this method requires authentication. If the requester is an administrator (normally only the case using the Server API):

1. One of cons_id, member_id, or primary_email must be specified in order to locate the record.
2. The order of precedence of the search terms is (a) cons_id (b) member_id, then (c) primary_email. If a record is not found or if multiple records are returned by any specified search term evaluated in this order, the method returns an error.

If the requester is a constituent using the Client API:

1. An authentication token representing the logged-in user must be passed.
2. Only the constituent record of the logged in user may be retrieved.

The structure of this method's response document may vary. The `<donation_detail>` element is only returned if `transaction_type=DONATION` donation transactions. It is omitted for other transaction_type values. Similarly, whether `<credit_card_>` or `<check_number>` elements are returned depends on what method of payment was used for the transaction.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

begin_date

Optional. The beginning of the date range of the transactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

end_date

Optional. The ending of the date range of the transactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

transaction_type

Optional. Specifies the type of transactions to return.

Type crm:transactionType.

Options are:

- DONATION
- PLEDGE
- RECURRING_GIFT

- TICKET
- DONATION_PENDING
- DONATION_PENDING_ACCEPTED
- DONATION_PENDING_REJECTED
- DONATION_PENDING_ACCEPTED_FAILED
- DONATION_PENDING_REJECTED_FAILED
- DONATION_ACCEPTED_FAILED
- DONATION_REJECTED_FAILED
- DONATION_SETTLED
- DONATION_REJECTED
- F2F_REGISTRATION
- TRIBUTES_REGISTRATION
- ECOMMERCE_PURCHASE
- REFUND
- FAILED_RECURRING_DONATION_DECLINED
- FAILED_RECURRING_DONATION_FRAUD

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

begin_date

Optional. The beginning of the date range of the transactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

end_date

Optional. The ending of the date range of the transactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

transaction_type

Optional. Specifies the type of transactions to return.

Type crm:transactionType.

Options are:

- DONATION
- PLEDGE
- RECURRING_GIFT
- TICKET
- DONATION_PENDING
- DONATION_PENDING_ACCEPTED
- DONATION_PENDING_REJECTED
- DONATION_PENDING_ACCEPTED_FAILED
- DONATION_PENDING_REJECTED_FAILED
- DONATION_ACCEPTED_FAILED
- DONATION_REJECTED_FAILED
- DONATION_SETTLED
- DONATION_REJECTED
- F2F_REGISTRATION
- TRIBUTES_REGISTRATION
- ECOMMERCE_PURCHASE
- REFUND
- FAILED_RECURRING_DONATION_DECLINED
- FAILED_RECURRING_DONATION_FRAUD

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getConsTransactionsResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <transaction>
    <timestamp>2013-02-03T05:18:27.499-06:00</timestamp>
    <confirmation_code/>
    <amount>
      <decimal>50.00</decimal>
      <formatted>$50.00</formatted>
    </amount>
    <tender_type>INVALID</tender_type>
    <pay_method>ONLINE</pay_method>
    <transaction_type>DONATION</transaction_type>
    <billing_name>
      <title xsi:nil="true"/>
      <first>John</first>
      <middle xsi:nil="true"/>
      <last>Smith</last>
      <suffix xsi:nil="true"/>
      <prof_suffix xsi:nil="true"/>
```

```
    </billing_name>
    <billing_address>
      <street1 xsi:nil="true"/>
      <street2 xsi:nil="true"/>
      <street3 xsi:nil="true"/>
      <city xsi:nil="true"/>
      <state xsi:nil="true"/>
      <zip xsi:nil="true"/>
      <county xsi:nil="true"/>
      <country xsi:nil="true"/>
    </billing_address>
    <donation_detail>
      <campaign>
        <id>9901</id>
        <title>API Campaign</title>
        <description>Sample campaign for API documentation.</description>
      </campaign>
      <form>
        <id>9999</id>
        <title>API Form</title>
        <description>Sample form for API documentation.</description>
      </form>
      <level>
        <id>9921</id>
        <title>API Donation Level</title>
      </level>
      <value_of_goods>
        <decimal>0.00</decimal>
        <formatted>$0.00</formatted>
      </value_of_goods>
      <tax_deductible_amount>
        <decimal>0.00</decimal>
        <formatted>$0.00</formatted>
      </tax_deductible_amount>
    </donation_detail>
  </transaction>
</getConsTransactionsResponse>
```

JSON response

```
{"getConsTransactionsResponse":{"transaction":{"amount":
{"formatted":"$50.00","decimal":"50.00"},"timestamp":"2013-02-03T05:18:27.563-06:00","pay
{"title":null,"last":"Smith","suffix":null,"middle":null,"first":"John","prof_suffix":nu
{},"donation_detail":{"value_of_goods":
{"formatted":"$0.00","decimal":"0.00"},"tax_deductible_amount":
{"formatted":"$0.00","decimal":"0.00"},"level":{"id":"9921","title":"API
 Donation Level"},"form":{"id":"9999","title":"API
 Form","description":"Sample form for API documentation."},"campaign":
{"id":"9901","title":"API Campaign","description":"Sample
 campaign for API documentation."}},"billing_address":
{"street2":null,"zip":null,"street1":null,"county":null,"state":null,"street3":null,"cou
```

isEmailValid

This method validates a supplied email address.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	GET, POST	Yes.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=isEmailValid &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&email=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]
Server API Syntax
`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=isEmailValid &api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] `&email=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

email

Required. Email address value to test for formal validity (method does not test whether the email address corresponds to an actual recipient).

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

email

Required. Email address value to test for formal validity (method does not test whether the email address corresponds to an actual recipient).

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<isEmailValidResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <valid>true</valid>
</isEmailValidResponse>
```

JSON response

```
{"isEmailValidResponse":{"valid":"true"}}
```

isUserInterested

This method checks to see if the specified constituent is associated with a given interest category.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=isUserInterested &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&interest_id=`*value* [`&cons_id=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=isUserInterested &api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] `&interest_id=`*value* [`&cons_id=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*]

Usage Notes

If called as a Client API, this method requires authentication. If the requester is an administrator (normally only the case using the Server API):

1. One of cons_id, member_id, or primary_email must be specified in order to locate the record.
2. The order of precedence of the search terms is (a) cons_id (b) member_id, then (c) primary_email. If a record is not found or if multiple records are returned by any specified search term evaluated in this order, the method returns an error.

If the requester is a constituent using the Client API:

1. An authentication token representing the logged-in user must be passed.
2. Only the constituent record of the logged in user may be retrieved.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

interest_id

Required. The numeric ID of the interest category to check.

Type xsd:nonNegativeInteger.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

interest_id

Required. The numeric ID of the interest category to check.

Type xsd:nonNegativeInteger.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<isConsInterestedResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <interested>true</interested>
</isConsInterestedResponse>
```

JSON response

```
{"isConsInterestedResponse":{"interested":"true"}}
```

isUserNameAvailable

Checks the availability of a username.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=isUserNameAvailable`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&user_name=`*value*

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=isUserNameAvailable`
`&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] `&user_name=`*value*

Usage Notes

The `isUserNameAvailable` method checks the availability of a username. A username is available if it has not already been taken by another user.

The user_name parameter must be a syntactically valid in order for the method to return a valid response.

A valid username can only be composed of letters, digits, '-', '_', '@', '.', '%', and ':'. By default, a username must be between 5 and 60 characters long. However, these values may be adjusted using the **USER_MIN_LOGIN_LEN** and **USER_MAX_LOGIN_LEN** SDPs.

If the username is not valid, an error response containing one of the following error codes will be issued:

- Error Code 9 - Request did not contain a required parameter. Missing parameter was: user_name
- Error Code 23 - Invalid length of User Name. User name must be between [minimum length] and [maximum length]
- Error Code 35 - Invalid characters in User Name (valid characters are letters, digits, '-', '_', '@', '.', '%', and ':')

In addition, this API will not check the availability of the user_name parameter if it is an email address. Instead, the following error code will be issued:

- Error Code 36 - Usernames that are email addresses are not checked.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

user_name

Required. The username that uniquely identifies a constituent

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

user_name

Required. The username that uniquely identifies a constituent

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<isUserNameAvailableResponse xmlns="http://convio.com/crm/v1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <available xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd">false</available>
</isUserNameAvailableResponse>
```

JSON response

```
{"isUserNameAvailableResponse":{"available":
{"content":"false","xsi:schemaLocation":"http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd"}}}
```

listCenters

Lists all centers that the caller has permission to view.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=listCenters`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=listCenters &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] [`& list_page_offset =`*value*] [`& list_page_size =`*value*] [`& list_record_offset =`*value*]

Usage Notes

If the Convio site is configured for Multi-Center mode, this method returns the list of Centers that the caller can access. Note that this may not include all centers defined for a particular Convio site, and may differ from one caller to another depending upon the caller's role and permissions.

The method supports paging of results.

If the Convio site is not configured for Multi-Center, this method returns an empty list.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listCentersResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <center>
    <id>1</id>
    <label>Center Name</label>
    <description>Center Description</description>
    <home_page>http://www.your_org.org/site/PageServer?pagename=center_home</
home_page>
    <not_found_page>http://www.your_org.org/site/PageServer?
pagename=center_not_found</not_found_page>
  </center>
</listCentersResponse>
```

JSON response

```
{"listCentersResponse":{"center":{"id":"1","description":"Center
 Description","label":"Center Name","home_page":"http://www.your_org.org/site/
PageServer?pagename=center_home","not_found_page":"http://www.your_org.org/
site/PageServer?pagename=center_not_found"}}}
```

listInteractions

This method can be used to get interactions for a specified interaction type. It will only return public custom interaction types. This method requires an interaction type ID.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=listInteractions`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&interaction_type_id=`*value* [`&begin_date=`*value*] [`&end_date=`*value*] [`&interaction_subject=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=listInteractions &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& interaction_type_id =`*value* [`& begin_date =`*value*] [`& end_date =`*value*] [`& interaction_subject =`*value*] [`& list_ascending =`*value*] [`& list_category_id =`*value*] [`& list_filter_column =`*value*] [`& list_filter_text =`*value*] [

`&list_page_offset=`*value*`][&list_page_size=`*value*`][&list_record_offset=`*value*`][&list_sort_column=`*value*`]`

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

interaction_type_id

Required. The Interaction Type to find. The type must be a public custom interactions.

Type xsd:nonNegativeInteger.

begin_date

Optional. The beginning of the date range of the interactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

end_date

Optional. The ending of the date range of the interactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

interaction_type_id

Required. The Interaction Type to find. The type must be a public custom interactions.

Type xsd:nonNegativeInteger.

begin_date

Optional. The beginning of the date range of the interactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

end_date

Optional. The ending of the date range of the interactions to return, formatted per the ISO 8601 specification.

Type xsd:dateTime.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listInteractionsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <interaction>
    <interactionId>1000</interactionId>
    <interactionTypeId>0</interactionTypeId>
    <categoryId>1000</categoryId>
    <referenceId>0</referenceId>
    <consId>1001001</consId>
    <createdBy>1001001</createdBy>
    <creationDate>2013-02-03T05:18:20.619-06:00</creationDate>
    <subject>Sample Subject 0</subject>
    <count>1</count>
    <note>
      <categoryId>1000</categoryId>
      <typeOfConsNote>normal</typeOfConsNote>
      <consId>1001001</consId>
      <createdBy>1001001</createdBy>
      <creationDate>2013-02-03T05:18:20.619-06:00</creationDate>
      <interactionId>1000</interactionId>
      <modifiedBy>1001001</modifiedBy>
      <modifiedDate>2013-02-03T05:18:20.619-06:00</modifiedDate>
      <noteId>1000</noteId>
      <subject>Sample Subject 0</subject>
      <text>Lorem ipsum dolor sit amet, consetetur sadipscing 0</text>
      <type>comment</type>
    </note>
  </interaction>
  <interaction>
    <interactionId>1001</interactionId>
    <interactionTypeId>1</interactionTypeId>
    <categoryId>1001</categoryId>
    <referenceId>0</referenceId>
    <consId>1001002</consId>
    <createdBy>1001002</createdBy>
    <creationDate>2013-02-03T05:18:20.619-06:00</creationDate>
    <subject>Sample Subject 1</subject>
```

```
    <count>2</count>
    <note>
      <categoryId>1001</categoryId>
      <typeOfConsNote>normal</typeOfConsNote>
      <consId>1001002</consId>
      <createdBy>1001002</createdBy>
      <creationDate>2013-02-03T05:18:20.619-06:00</creationDate>
      <interactionId>1001</interactionId>
      <modifiedBy>1001002</modifiedBy>
      <modifiedDate>2013-02-03T05:18:20.619-06:00</modifiedDate>
      <noteId>1001</noteId>
      <subject>Sample Subject 1</subject>
      <text>Lorem ipsum dolor sit amet, consetetur sadipscing 1</text>
      <type>comment</type>
    </note>
  </interaction>
  <interaction>
    <interactionId>1002</interactionId>
    <interactionTypeId>2</interactionTypeId>
    <categoryId>1002</categoryId>
    <referenceId>0</referenceId>
    <consId>1001003</consId>
    <createdBy>1001003</createdBy>
    <creationDate>2013-02-03T05:18:20.619-06:00</creationDate>
    <subject>Sample Subject 2</subject>
    <count>3</count>
    <note>
      <categoryId>1002</categoryId>
      <typeOfConsNote>normal</typeOfConsNote>
      <consId>1001003</consId>
      <createdBy>1001003</createdBy>
      <creationDate>2013-02-03T05:18:20.619-06:00</creationDate>
      <interactionId>1002</interactionId>
      <modifiedBy>1001003</modifiedBy>
      <modifiedDate>2013-02-03T05:18:20.619-06:00</modifiedDate>
      <noteId>1002</noteId>
      <subject>Sample Subject 2</subject>
      <text>Lorem ipsum dolor sit amet, consetetur sadipscing 2</text>
      <type>comment</type>
    </note>
  </interaction>
</listInteractionsResponse>
```

JSON response

```
{"listInteractionsResponse":{"interaction":
[{"creationDate":"2013-02-03T05:18:20.619-06:00","interactionTypeId":"0","count":"1","cre
 Subject 0","categoryId":"1000","referenceId":"0","note":
{"creationDate":"2013-02-03T05:18:20.619-06:00","text":"Lorem ipsum dolor sit
 amet, consetetur sadipscing
 0","interactionId":"1000","createdBy":"1001001","consId":"1001001","typeOfConsNote":"nor
 Subject
 0","modifiedBy":"1001001","categoryId":"1000","noteId":"1000","type":"comment","modifiec
{"creationDate":"2013-02-03T05:18:20.619-06:00","interactionTypeId":"1","count":"2","crea
 Subject 1","categoryId":"1001","referenceId":"0","note":
{"creationDate":"2013-02-03T05:18:20.619-06:00","text":"Lorem ipsum dolor sit
 amet, consetetur sadipscing
 1","interactionId":"1001","createdBy":"1001002","consId":"1001002","typeOfConsNote":"nor
 Subject
 1","modifiedBy":"1001002","categoryId":"1001","noteId":"1001","type":"comment","modifiec
{"creationDate":"2013-02-03T05:18:20.619-06:00","interactionTypeId":"2","count":"3","crea
```

```
 Subject 2","categoryId":"1002","referenceId":"0","note":
{"creationDate":"2013-02-03T05:18:20.619-06:00","text":"Lorem ipsum dolor sit
 amet, consetetur sadipscing
 2","interactionId":"1002","createdBy":"1001003","consId":"1001003","typeOfConsNote":"no
 Subject
 2","modifiedBy":"1001003","categoryId":"1002","noteId":"1002","type":"comment","modifie
```


listInterests

This method returns the list of all interest categories, optionally including interests assigned to security categories normally not visible to the caller.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=listInterests`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&cons_id=`*value*] [`&include_hidden=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=listInterests &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] [`& cons_id =`*value*] [`& include_hidden =`*value*] [`& member_id =`*value*] [`& primary_email =`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

include_hidden

Optional. Determines whether to include hidden interests (interests in security categories normally not visible to the caller) in the result set.

Type xsd:boolean.

Default is false.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

include_hidden

Optional. Determines whether to include hidden interests (interests in security categories normally not visible to the caller) in the result set.

Type xsd:boolean.

Default is false.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listInterestsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <interest>
    <id>1001</id>
    <label>Interest Name</label>
    <path>Interest Full Name</path>
    <for_email>false</for_email>
    <for_web>true</for_web>
    <parent>Parent Category</parent>
    <description>Interest Description</description>
    <center_id>1000</center_id>
    <center_label>Center Name</center_label>
  </interest>
</listInterestsResponse>
```

JSON response

```
{"listInterestsResponse":{"interest":{"id":"1001","center_label":"Center
 Name","description":"Interest Description","for_web":"true","path":"Interest
 Full Name","parent":"Parent Category","for_email":"false","label":"Interest
 Name","center_id":"1000"}}}
```

listUserFieldChoices

This method returns a comma-separated list of the valid values for a given field in the constituent record, if a list of valid values is defined. For fields which do not have a range of predefined values, this method returns an empty result set. For Date attributes, this method returns a list of valid Years.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=listUserFieldChoices &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&field=`*value*

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=listUserFieldChoices & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& field =`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

field

Required. Specifies the name of the constituent field for which to return the possible values.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

field

Required. Specifies the name of the constituent field for which to return the possible values.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listConsFieldChoicesResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <choice>Mr.</choice>
  <choice>Ms.</choice>
  <choice>Mrs.</choice>
  <choice>Miss</choice>
  <choice>Dr.</choice>
</listConsFieldChoicesResponse>
```

JSON response

```
{"listConsFieldChoicesResponse":{"choice":["Mr.","Ms.","Mrs.","Miss","Dr."]}}
```

listUserFields

This method returns the list of fields in the constituent record that can be viewed or changed by the caller. If include_choices is true, then it will return choices with each field when they are available. If sort_order is "group", then it will return field sorted by group information.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=listUserFields`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&access=`*view* | *update*] [`&include_choices=`*value*] [`&sort_order=`*field* | *group*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=listUserFields &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] [`& access =`*view* | *update*] [`& include_choices =`*value*] [`& sort_order =`*field* | *group*]

Usage Notes

This method provides detailed information about each of the fields available to the calling user. The elements within each field element of the response document include:

name	This is the name of the HTTP parameter to use to set the field in a request to create or update a constituent. Dots in the name reflect the structure of nested elements in the response to a request to get a constituent's record.
label	This is what the Convio system uses as a field label when presenting the field in its UI.
valueType	Indicates the type of data in the field. See the list below for possible values.
maxChars	For "TEXT" type fields, this indicates the maximum number of characters that can be stored.

The possible values for the valueType element are:

BOOLEAN	These fields have a value of "true" or "false".
DATE	These fields have a date and time value formatted as defined by the XML Schema specification, which is derived from the ISO 8601 specification. This format has a date part, a time part, and a time zone part. If the time zone part is omitted, then the time zone configured for the site is used. If the time part is omitted, then the time is set to the beginning of the day (00:00:00) in the time zone used.
ENUMERATION	These fields have a finite set of possible values which are specified in the XML Schema document.

MONEY	These fields have monetary values that are formatted as currency using the site's configured locale information.
NUMBER	These fields have integer values with no formatting (no punctuation such as commas or points).
TEXT	These fields have text strings.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

access

Optional. Specifies the type of constituent record access for which to list the available fields. If the value is "view" then the response includes fields that the caller can view. If the value is "update" then the response includes fields that the caller can change.

Type xsd:string.

Default is view.

Options are:

- view
- update

include_choices

Optional. The flag to indicate whether or not each textfield contains choice information.

Type xsd:boolean.

Default is false.

sort_order

Optional. Specifies the order of the records. If the value is "field", then response will be sorted alphabetically in field name. If the value is "group", then response will be sorted by group.

Type xsd:string.

Default is field.

Options are:

- field
- group

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

access

Optional. Specifies the type of constituent record access for which to list the available fields. If the value is "view" then the response includes fields that the caller can view. If the value is "update" then the response includes fields that the caller can change.

Type xsd:string.

Default is view.

Options are:

- view
- update

include_choices

Optional. The flag to indicate whether or not each textfield contains choice information.

Type xsd:boolean.

Default is false.

sort_order

Optional. Specifies the order of the records. If the value is "field", then response will be sorted alphabetically in field name. If the value is "group", then response will be sorted by group.

Type xsd:string.

Default is field.

Options are:

- field
- group

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listConsFieldsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <field>
    <label>Text Field Label</label>
    <name>text_field</name>
    <required>true</required>
    <maxChars>255</maxChars>
    <choices>
      <choice>Choice Label One</choice>
      <choice>Choice Label Two</choice>
    </choices>
    <group>Group Label</group>
    <groupOrder>3</groupOrder>
    <subGroup>Sub Group Label</subGroup>
    <subGroupOrder>1</subGroupOrder>
    <valueType>TEXT</valueType>
    <isSensitive>false</isSensitive>
    <requiredForUser>true</requiredForUser>
    <isUserModifiable>true</isUserModifiable>
  </field>
  <field>
    <label>Number Field Label</label>
    <name>number_field</name>
    <required>false</required>
    <group>Group Label</group>
    <subGroup>Sub Group Label</subGroup>
    <groupOrder>2</groupOrder>
    <valueType>NUMBER</valueType>
    <isSensitive>false</isSensitive>
    <requiredForUser>false</requiredForUser>
    <isUserModifiable>true</isUserModifiable>
  </field>
  <field>
    <label>Boolean Field Label</label>
    <name>boolean_field</name>
    <required>false</required>
    <valueType>BOOLEAN</valueType>
    <isSensitive>false</isSensitive>
```

```
    <requiredForUser>false</requiredForUser>
    <isUserModifiable>true</isUserModifiable>
  </field>
  <field>
    <label>Enum Field Label</label>
    <name>enum_field</name>
    <required>false</required>
    <valueType>ENUMERATION</valueType>
    <isSensitive>false</isSensitive>
    <requiredForUser>false</requiredForUser>
    <isUserModifiable>false</isUserModifiable>
  </field>
  <field>
    <label>Money Field Label</label>
    <name>money_field</name>
    <required>false</required>
    <valueType>MONEY</valueType>
    <isSensitive>false</isSensitive>
    <requiredForUser>false</requiredForUser>
    <isUserModifiable>false</isUserModifiable>
  </field>
  <field>
    <label>Date Field Label</label>
    <name>date_field</name>
    <required>false</required>
    <valueType>DATE</valueType>
    <isSensitive>false</isSensitive>
    <requiredForUser>false</requiredForUser>
    <isUserModifiable>true</isUserModifiable>
  </field>
</listConsFieldsResponse>
```

JSON response

```
{"listConsFieldsResponse":{"field":
[{"isUserModifiable":"true","maxChars":"255","choices":{"choice":
["Choice Label One","Choice Label Two"]},"label":"Text Field
 Label","requiredForUser":"true","name":"text_field","groupOrder":"3","subGroupOrder":"1"
 Group
 Label","valueType":"TEXT","isSensitive":"false","required":"true","group":"Group
 Label"},
{"isUserModifiable":"true","requiredForUser":"false","groupOrder":"2","name":"number_fie
 Group Label","label":"Number Field
 Label","isSensitive":"false","group":"Group Label","required":"false"},
{"isUserModifiable":"true","requiredForUser":"false","name":"boolean_field","valueType":"
 Field Label","isSensitive":"false","required":"false"},
{"isUserModifiable":"false","requiredForUser":"false","name":"enum_field","valueType":"EN
 Field Label","isSensitive":"false","required":"false"},
{"isUserModifiable":"false","requiredForUser":"false","name":"money_field","valueType":"M
 Field Label","isSensitive":"false","required":"false"},
{"isUserModifiable":"true","requiredForUser":"false","name":"date_field","valueType":"DAT
 Field Label","isSensitive":"false","required":"false"}]}}
```

logInteraction

This method can be used to manually log an interaction record for the specified constituent containing information supplied by the caller. It must be called by a logged-in user within an authenticated session. This method requires at least one unique identifier for the user (may be cons_id, member_id, or primary_email). A custom interaction type ID may be specified. If it is not then then the interaction logged is of type "Other".The body of the comment has a maximum size of 1500 characters.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=logInteraction`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&interaction_subject=`*value* [`&cons_id=`*value*] [`&interaction_body=`*value*] [`&interaction_cat_id=`*value*] [`&interaction_count=`*value*] [`&interaction_type_id=`*value*] [`&member_id=`*value*] [`&primary_email=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=logInteraction &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& interaction_subject =`*value* [`& cons_id =`*value*] [`& interaction_body =`*value*] [`& interaction_cat_id =`*value*] [`& interaction_count =`*value*] [`& interaction_type_id =`*value*] [`& member_id =`*value*] [`& primary_email =`*value*]

Usage Notes

If called as a Client API, this method requires authentication. If the requester is an administrator (normally only the case using the Server API):

1. One of cons_id, member_id, or primary_email must be specified in order to locate the record.
2. The order of precedence of the search terms is (a) cons_id (b) member_id, then (c) primary_email. If a record is not found or if multiple records are returned by any specified search term evaluated in this order, the method returns an error.

If the requester is a constituent using the Client API:

1. An authentication token representing the logged-in user must be passed.
2. An interaction is recorded for the logged in user.

This method is automatically invoked by any call to the `create`, or by any call to `update` or `createOrUpdate` that specifies one of the unique identifier parameters listed above.

By default, the `logInteraction` method creates an interaction record for the constituent of type 0='Other,' using the values (if any) supplied in the interaction_cat_id, interaction_subject, interaction_body, and interaction_count parameters, or their defaults. If the interaction_type_id is specified with a valid, custom interaction type ID an interaction of that type will be created. The method also logs the caller's IP address and the user account ID for the caller's login session (may be either the constituent's ID or the ID of an administrator account used by the caller.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

interaction_subject

Required. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

interaction_body

Optional. Text describing or associated with the interaction.

Type xsd:string.

interaction_cat_id

Optional. The Security Category associated with this interaction. The default is 0, which corresponds to the "General" category.

Type xsd:nonNegativeInteger.

Default is 0.

interaction_count

Optional. The number of times the constituent performed this interaction.

Type xsd:nonNegativeInteger.

Default is 1.

interaction_type_id

Optional. The Interaction Type associated with this interaction. If an Interaction Type is specified it must be a custom interaction.The default is 0, which corresponds to the "Other" Interaction Type.

Type xsd:nonNegativeInteger.

Default is 0.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

interaction_subject

Required. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

interaction_body

Optional. Text describing or associated with the interaction.

Type xsd:string.

interaction_cat_id

Optional. The Security Category associated with this interaction. The default is 0, which corresponds to the "General" category.

Type xsd:nonNegativeInteger.

Default is 0.

interaction_count

Optional. The number of times the constituent performed this interaction.

Type xsd:nonNegativeInteger.

Default is 1.

interaction_type_id

Optional. The Interaction Type associated with this interaction. If an Interaction Type is specified it must be a custom interaction.The default is 0, which corresponds to the "Other" Interaction Type.

Type xsd:nonNegativeInteger.

Default is 0.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateConsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cons_id>1000121</cons_id>
  <message>Interaction logged successfully.</message>
  <interaction_id>1001</interaction_id>
</updateConsResponse>
```

JSON response

```
{"updateConsResponse":{"message":"Interaction logged
 successfully.","cons_id":"1000121","interaction_id":"1001"}}
```

update

This method updates an existing user record and returns the record's unique identifier. The list of fields that can be updated is customizable for each organization and is different for administrative and non-administrative users. The fields that can be updated by the caller can be retrieved using the listUserFields API method.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=update &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&add_center_ids=`*value*

][&add_center_opt_in_ids=*value*][&add_group_ids=*value*][&add_interest_ids=*value*][&cons_id=*value*][&interaction_body=*value*][&interaction_cat_id=*value*][&interaction_count=*value*][&interaction_subject=*value*][&member_id=*value*][&primary_email=*value*][&remove_center_ids=*value*][&remove_center_opt_in_ids=*value*][&remove_group_ids=*value*][&remove_interest_ids=*value*][&suppress_cleaning=*value*][&auth=*value*][&sso_auth_token=*value*]

Server API Syntax

https://*secure2.convio.net/organization*/site/SRConsAPI?method=update & api_key =*value* & login_name =*value* & login_password =*value* & v =*value* [& center_id =*value*][& response_format =*xml* |*json*][& source =*value*][& sub_source =*value*][& suppress_response_codes =*value*][& add_center_ids =*value*][& add_center_opt_in_ids =*value*][& add_group_ids =*value*][& add_interest_ids =*value*][& cons_id =*value*][& interaction_body =*value*][& interaction_cat_id =*value*][& interaction_count =*value*][& interaction_subject =*value*][& member_id =*value*][& primary_email =*value*][& remove_center_ids =*value*][& remove_center_opt_in_ids =*value*][& remove_group_ids =*value*][& remove_interest_ids =*value*][& suppress_cleaning =*value*]

Usage Notes

This method attempts to locate and update an existing constituent record in the Convio Constituent 360 database using the value of the cons_id, member_id, or primary_email parameters to locate the record.

If called as a Client API, this method requires authentication. If the requester is an administrator (normally only the case using the Server API):

1. One of cons_id, member_id, or primary_email must be specified in order to locate the record to update.
2. The order of precedence of the search terms is (a) cons_id (b) member_id, then (c) primary_email. If a record is not found or if multiple records are returned by any specified search term evaluated in this order, the method returns an error.
3. If a request specifies more than one search parameter, then the system will attempt to update the values of lower-precedence parameters using the values provided. This may result in an error if a duplicate value exists in the database for a specified member_id or primary_email.

If the requester is a constituent using the Client API:

1. An authentication token representing the logged-in user must be passed.
2. If the logged-in user is not an administrator, only the constituent record of the logged in user may be updated.

The list of constituent record fields that can be set is customizable for each organization. It is different for administrative users (via the Client or Server API) and non-administrative users (via Client API).

listUserFields will return the list of the Constituent record fields the caller may update.

This method automatically invokes *logInteraction* to log an interaction for the target constituent.

Note: This method will not allow you to add the created constituent directly to an Interest Opt-In or Opt-Out group. You should instead add or remove the interest to the constituent, which will adjust the group membership as appropriate.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

add_center_ids

Optional. Comma-separated list of numeric IDs of centers to add the user to.

Type xsd:string.

add_center_opt_in_ids

Optional. Comma-separated list of numeric IDs of centers for which the user should be opted in to email communications.

Type xsd:string.

add_group_ids

Optional. Comma-separated list of numeric IDs of groups to add the user to. Only available when called using an administrator account. Cannot be used for pre-defined groups (those with IDs less than 1000). Cannot be used for administrator security groups.

Type xsd:string.

add_interest_ids

Optional. Comma-separated list of numeric IDs of email interests to add the user to.

Type xsd:string.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

interaction_body

Optional. Text describing or associated with the interaction.

Type xsd:string.

interaction_cat_id

Optional. The Security Category associated with this interaction. The default is 0, which corresponds to the "General" category.

Type xsd:nonNegativeInteger.

Default is 0.

interaction_count

Optional. The number of times the constituent performed this interaction.

Type xsd:nonNegativeInteger.

Default is 1.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent. Required to create a new constituent record. Optional when updating a record.

Type xsd:string.

remove_center_ids

Optional. Comma-separated list of numeric IDs of centers to remove the user from.

Type xsd:string.

remove_center_opt_in_ids

Optional. Comma-separated list of numeric IDs of centers for which the user should be opted out of email communications.

Type xsd:string.

remove_group_ids

Optional. Comma-separated list of numeric IDs of groups to remove the user from. Only valid when updating records. Only available when called using an administrator account. Cannot be used for pre-defined groups (those with IDs less than 1000). Cannot be used for administrator security groups.

Type xsd:string.

remove_interest_ids

Optional. Comma-separated list of numeric IDs of email interests to remove the user from. Only valid when updating records.

Type xsd:string.

suppress_cleaning

Optional. Specifies whether data cleaning/standardization should be turned on or off. This parameter is intended for performance tuning for server side batch APIs calls.

Type xsd:boolean.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

add_center_ids

Optional. Comma-separated list of numeric IDs of centers to add the user to.

Type xsd:string.

add_center_opt_in_ids

Optional. Comma-separated list of numeric IDs of centers for which the user should be opted in to email communications.

Type xsd:string.

add_group_ids

Optional. Comma-separated list of numeric IDs of groups to add the user to. Only available when called using an administrator account. Cannot be used for pre-defined groups (those with IDs less than 1000). Cannot be used for administrator security groups.

Type xsd:string.

add_interest_ids

Optional. Comma-separated list of numeric IDs of email interests to add the user to.

Type xsd:string.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

interaction_body

Optional. Text describing or associated with the interaction.

Type xsd:string.

interaction_cat_id

Optional. The Security Category associated with this interaction. The default is 0, which corresponds to the "General" category.

Type xsd:nonNegativeInteger.

Default is 0.

interaction_count

Optional. The number of times the constituent performed this interaction.

Type xsd:nonNegativeInteger.

Default is 1.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent. Required to create a new constituent record. Optional when updating a record.

Type xsd:string.

remove_center_ids

Optional. Comma-separated list of numeric IDs of centers to remove the user from.

Type xsd:string.

remove_center_opt_in_ids

Optional. Comma-separated list of numeric IDs of centers for which the user should be opted out of email communications.

Type xsd:string.

remove_group_ids

Optional. Comma-separated list of numeric IDs of groups to remove the user from. Only valid when updating records. Only available when called using an administrator account. Cannot be used for pre-defined groups (those with IDs less than 1000). Cannot be used for administrator security groups.

Type xsd:string.

remove_interest_ids

Optional. Comma-separated list of numeric IDs of email interests to remove the user from. Only valid when updating records.

Type xsd:string.

suppress_cleaning

Optional. Specifies whether data cleaning/standardization should be turned on or off. This parameter is intended for performance tuning for server side batch APIs calls.

Type xsd:boolean.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateConsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cons_id>1001002</cons_id>
  <message>User updated.</message>
</updateConsResponse>
```

JSON response

```
{"updateConsResponse":{"message":"User updated.","cons_id":"1001002"}}
```

updateInteraction

This method can be used to update an interaction record with information supplied by the caller. It must be called by a logged-in user within an authenticated session. This method requires an interaction ID. The interaction being updated can only be of the interaction type "Other" or a custom interaction type.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=updateInteraction &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&interaction_id=`*value* [`&interaction_body=`*value*] [`&interaction_cat_id=`*value*] [`&interaction_count=`*value*] [`&interaction_subject=`*value*] [`&interaction_type_id=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=updateInteraction & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& interaction_id =`*value* [`& interaction_body =`*value*] [`& interaction_cat_id =`*value*] [`& interaction_count =`*value*] [`& interaction_subject =`*value*] [`& interaction_type_id =`*value*]

Usage Notes

If called as a Client API, this method requires authentication. If the requester is a constituent using the Client API an authentication token representing the logged-in user must be passed.

The `updateInteraction` method updates an existing interaction record for the constituent using the values (if any) supplied in the interaction_cat_id, interaction_subject, interaction_body, interaction_count and interaction_id parameters. Only interactions of type 0='Other' or a custom interaction type can be updated.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

interaction_id

Required. The id of the interaction to update.

Type xsd:nonNegativeInteger.

interaction_body

Optional. Text describing or associated with the interaction.

Type xsd:string.

interaction_cat_id

Optional. The Security Category associated with this interaction. The default is 0, which corresponds to the "General" category.

Type xsd:nonNegativeInteger.

Default is 0.

interaction_count

Optional. The number of times the constituent performed this interaction.

Type xsd:nonNegativeInteger.

Default is 1.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

interaction_type_id

Optional. The Interaction Type associated with this interaction.

Type xsd:nonNegativeInteger.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

interaction_id

Required. The id of the interaction to update.

Type xsd:nonNegativeInteger.

interaction_body

Optional. Text describing or associated with the interaction.

Type xsd:string.

interaction_cat_id

Optional. The Security Category associated with this interaction. The default is 0, which corresponds to the "General" category.

Type xsd:nonNegativeInteger.

Default is 0.

interaction_count

Optional. The number of times the constituent performed this interaction.

Type xsd:nonNegativeInteger.

Default is 1.

interaction_subject

Optional. Short, descriptive subject of the user interaction. Can be up to 80 characters in length.

Type xsd:string.

interaction_type_id

Optional. The Interaction Type associated with this interaction.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateInteractionResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cons_id>1001001</cons_id>
  <message>Interaction updated successfully.</message>
  <interactionId>1001</interactionId>
</updateInteractionResponse>
```

JSON response

```
{"updateInteractionResponse":{"message":"Interaction updated
 successfully.","interactionId":"1001","cons_id":"1001001"}}
```

Content API

CRContentAPI
`https://secure2.convio.net/organization/site/CRContentAPI`

Description

The Content API provides integration points for Convio content types.

API Administrator accounts used for Server API calls to this API must belong to an Administrative Security Group that has "Use Convio Content API" permission in the "Content - All Authoring" permission type. The built-in "API Administrators" group has this permission by default.

listLinkProviders

Returns a list of link providers, each of which supplies links to a type of objects in the site.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRContentAPI	SRContentAPI	GET, POST	Server.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRContentAPI?method=listLinkProviders`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&for_email=`*value*]

Server API Syntax
`https://`*secure2.convio.net/organization*`/site/SRContentAPI?method=listLinkProviders`
`&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id`

=*value*] [& `response_format` =*xml* |*json*] [& `source` =*value*] [& `sub_source` =*value*] [& `suppress_response_codes` =*value*] [& `for_email` =*value*]

Usage Notes

This method lists Link Providers that aggregate links to different types of content within Convio system. Use this API to select a link provider ID required by the *listLinks* method.

The for_email parameter limits the list of providers returned to those that are suitable for embedding in the body of outgoing email messages.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

for_email

Optional. Specifies whether the text being edited is intended for display in an email message: "true" or "false". Some types of links are only applicable, or are not applicable, within email messages.

Type xsd:boolean.

Default is true.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

for_email

Optional. Specifies whether the text being edited is intended for display in an email message: "true" or "false". Some types of links are only applicable, or are not applicable, within email messages.

Type xsd:boolean.

Default is true.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listLinkProvidersResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <provider>
    <label>Link Type A</label>
    <typeId>42</typeId>
  </provider>
  <provider>
    <label>Link Type B</label>
    <typeId>43</typeId>
  </provider>
</listLinkProvidersResponse>
```

JSON response

```
{"listLinkProvidersResponse":{"provider":[{"label":"Link Type
 A","typeId":"42"},{"label":"Link Type B","typeId":"43"}]}}
```

listLinks

Returns a list of links available for a given link provider.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRContentAPI	SRContentAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRContentAPI?method=listLinks`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&provider_id=`*value* [`&clear_cache=`*value*] [`&for_email=`*value*] [`&list_ascending=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*displayName* | *editName* | *description* | *id*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRContentAPI?method=listLinks &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& provider_id =`*value* [`& clear_cache =`*value*] [`& for_email =`*value*] [`& list_ascending =`*value*] [`& list_filter_text =`*value*] [`& list_page_offset =`*value*] [`& list_page_size =`*value*] [`& list_record_offset =`*value*] [`& list_sort_column =`*displayName* | *editName* | *description* | *id*]

Usage Notes

Given a provider_id, this method returns a list of links supplied by that provider to content within the Convio system. Use *listLinkProviders* to return the list of valid provider_id values.

The for_email parameter limits the list of links returned to those that are suitable for embedding in the body of outgoing email messages.

Client API callers must be logged in and must be administrator users with appropriate Content Authoring rights. Server API callers must have "Content - All Authoring/Use Convio Content API" permission.

Note: The links available during a session are cached. If the list of valid links changes during the session, (i.e. if content is added or deleted) use clear_cache to force the cached list of links to refresh.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

provider_id

Required. The ID of the provider of links.

Type xsd:integer.

clear_cache

Optional. Indicates whether to clear cached results from prior invocations. For better performance when paging through large result sets, results are kept in memory while paging. Use this parameter if you need the cache to be reset.

Type xsd:boolean.

Default is false.

for_email

Optional. Specifies whether the text being edited is intended for display in an email message: "true" or "false". Some types of links are only applicable, or are not applicable, within email messages.

Type xsd:boolean.

Default is true.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Options are:

- displayName
- editName
- description
- id

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

provider_id

Required. The ID of the provider of links.

Type xsd:integer.

clear_cache

Optional. Indicates whether to clear cached results from prior invocations. For better performance when paging through large result sets, results are kept in memory while paging. Use this parameter if you need the cache to be reset.

Type xsd:boolean.

Default is false.

for_email

Optional. Specifies whether the text being edited is intended for display in an email message: "true" or "false". Some types of links are only applicable, or are not applicable, within email messages.

Type xsd:boolean.

Default is true.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Options are:

- displayName
- editName
- description
- id

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listLinksResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <pagingMetadata>
    <currentSize>2</currentSize>
    <isFinalSize>true</isFinalSize>
  </pagingMetadata>
```

```
  <linkableObject>
    <id>12</id>
    <displayName>Link Alpha</displayName>
    <editName>Edit Link Alpha</editName>
    <description>Link Alpha Description</description>
    <url>http://example.com/InsertExample?opt1=alpah&amp;opt2=2</url>
    <optionsAllowed>true</optionsAllowed>
    <autoLoginAllowed>true</autoLoginAllowed>
    <pageList>
      <page>
        <label>Page 1</label>
        <url>http://example.com/InsertExample?
opt1=alpah&amp;opt2=2&amp;page=p1</url>
        <linkParameter>
          <label>Optional parameter for page 1 for this target.</label>
          <name>page_option1</name>
          <choice>
            <value>thing1</value>
            <label>First value</label>
          </choice>
          <choice>
            <value>thing2</value>
            <label>Second value</label>
          </choice>
        </linkParameter>
        <linkParameter>
          <label>Another parameter for page 1 for this target.</label>
          <name>page_option2</name>
          <choice>
            <value>item1</value>
            <label>Some value</label>
          </choice>
          <choice>
            <value>item2</value>
            <label>Another value</label>
          </choice>
        </linkParameter>
      </page>
    </pageList>
    <linkParameter>
      <label>Optional parameter for all pages for this target.</label>
      <name>option1</name>
      <choice>
        <value>value1</value>
        <label>Value 1</label>
      </choice>
      <choice>
        <value>value2</value>
        <label>Value 2</label>
      </choice>
    </linkParameter>
  </linkableObject>
  <linkableObject>
    <id>14</id>
    <displayName>Link Bravo</displayName>
    <editName>Edit Link Bravo</editName>
    <description>Link Bravo Description.  Bravo has no options.</description>
    <url>http://example.com/InsertExample?opt1=bravo&amp;opt2=2</url>
    <optionsAllowed>false</optionsAllowed>
    <autoLoginAllowed>true</autoLoginAllowed>
  </linkableObject>
</listLinksResponse>
```

JSON response


```
{"listLinksResponse":{"linkableObject":
[{"optionsAllowed":"true","id":"12","pageList":{"page":
{"linkParameter":[{"choice":[{"value":"thing1","label":"First value"},
{"value":"thing2","label":"Second
 value"}],"name":"page_option1","label":"Optional
 parameter for page 1 for this target."},{"choice":
[{"value":"item1","label":"Some value"},{"value":"item2","label":"Another
 value"}],"name":"page_option2","label":"Another parameter for page 1 for
 this target."}],"label":"Page 1","url":"http://example.com/InsertExample?
opt1=alpah&opt2=2&page=p1"}},"autoLoginAllowed":"true","editName":"Edit Link
 Alpha","linkParameter":{"choice":[{"value":"value1","label":"Value 1"},
{"value":"value2","label":"Value 2"}],"name":"option1","label":"Optional
 parameter for all pages for this target."},"description":"Link
 Alpha Description","displayName":"Link Alpha","url":"http://
example.com/InsertExample?opt1=alpah&opt2=2"},
{"optionsAllowed":"false","id":"14","autoLoginAllowed":"true","editName":"Edit
 Link Bravo","description":"Link Bravo Description.  Bravo
 has no options.","displayName":"Link Bravo","url":"http://
example.com/InsertExample?opt1=bravo&opt2=2"}],"pagingMetadata":
{"isFinalSize":"true","currentSize":"2"}}}
```


listImages

Returns a list of images from the Image Library.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRContentAPI	SRContentAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRContentAPI?method=listImages`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&include_archived=`*true* | *false*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*creation_date*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRContentAPI?method=listImages &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] [`& include_archived =`*true* | *false*] [`& list_ascending =`*value*] [`& list_category_id =`*value*] [`& list_filter_text =`*value*] [`& list_page_offset =`*value*] [`& list_page_size =`*value*] [`& list_record_offset =`*value*] [`& list_sort_column =`*creation_date*]

Usage Notes

This method returns a list of links to image content stored on the Convio site.

The list_filter_text parameter specifies search text to filter the results. It matches, case insensitively, text in the file name, title, or keywords of images.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

include_archived

Optional. Specifies whether to include archived images in the results.

Type xsd:boolean.

Default is false.

Options are:

- true
- false

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Options are:

- creation_date

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

include_archived

Optional. Specifies whether to include archived images in the results.

Type xsd:boolean.

Default is false.

Options are:

- true
- false

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Options are:

- creation_date

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listImagesResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <pagingMetadata>
    <currentSize>2</currentSize>
    <isFinalSize>true</isFinalSize>
  </pagingMetadata>
  <image>
    <imageId>1000</imageId>
```

```
    <title>Sample image 1000</title>
    <createdDate>2013-02-03T05:18:24.118-06:00</createdDate>
    <originalFilename>local_image.jpg</originalFilename>
    <keywords>convio, logo</keywords>
    <insecureUrl>http://www.convio.com/images/content/image_1000.jpg</
insecureUrl>
    <secureUrl>https://www.convio.com/images/content/image_1000.jpg</
secureUrl>
    <categoryId>0</categoryId>
    <archived>false</archived>
    <height>0</height>
    <width>0</width>
  </image>
  <image>
    <imageId>1001</imageId>
    <title>Sample image 1001</title>
    <createdDate>2013-02-03T05:18:24.120-06:00</createdDate>
    <originalFilename>local_image.jpg</originalFilename>
    <keywords>convio, logo</keywords>
    <insecureUrl>http://www.convio.com/images/content/image_1001.jpg</
insecureUrl>
    <secureUrl>https://www.convio.com/images/content/image_1001.jpg</
secureUrl>
    <categoryId>0</categoryId>
    <archived>false</archived>
    <height>0</height>
    <width>0</width>
  </image>
  <showSecurityCatFilter>true</showSecurityCatFilter>
</listImagesResponse>
```

JSON response

```
{"listImagesResponse":{"pagingMetadata":
{"isFinalSize":"true","currentSize":"2"},"showSecurityCatFilter":"true","image":
[{"title":"Sample image 1002","height":"0","keywords":"convio,
 logo","imageId":"1002","archived":"false","width":"0","categoryId":"0","insecureUrl":"ht
www.convio.com/images/content/
image_1002.jpg","originalFilename":"local_image.jpg","secureUrl":"https://
www.convio.com/images/content/
image_1002.jpg","createdDate":"2013-02-03T05:18:24.121-06:00"},
{"title":"Sample image 1003","height":"0","keywords":"convio,
 logo","imageId":"1003","archived":"false","width":"0","categoryId":"0","insecureUrl":"ht
www.convio.com/images/content/
image_1003.jpg","originalFilename":"local_image.jpg","secureUrl":"https://
www.convio.com/images/content/
image_1003.jpg","createdDate":"2013-02-03T05:18:24.121-06:00"}]}}
```

listSupportedLocales

Supports the retrieval of all locales supported by the site.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRContentAPI	SRContentAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRContentAPI?method=listSupportedLocales`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [

`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRContentAPI?method=listSupportedLocales` `&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*]

Usage Notes

This method returns a list of the locales supported by the Convio site.

No method-specific parameters are supported by this method.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listSupportedLocalesResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <supportedLocale>
    <fullyQualifiedName>en_US</fullyQualifiedName>
    <displayName>English (United States)</displayName>
    <defaultLocale>true</defaultLocale>
    <country>US</country>
    <displayCountry>United States</displayCountry>
    <language>en</language>
    <displayLanguage>English</displayLanguage>
    <variant/>
    <displayVariant/>
  </supportedLocale>
  <supportedLocale>
    <fullyQualifiedName>en_GB</fullyQualifiedName>
    <displayName>English (United Kingdom)</displayName>
    <defaultLocale>false</defaultLocale>
    <country>GB</country>
    <displayCountry>United Kingdom</displayCountry>
    <language>en</language>
    <displayLanguage>English</displayLanguage>
    <variant/>
    <displayVariant/>
  </supportedLocale>
  <supportedLocale>
    <fullyQualifiedName>en_CA</fullyQualifiedName>
    <displayName>English (Canada)</displayName>
    <defaultLocale>false</defaultLocale>
    <country>CA</country>
    <displayCountry>Canada</displayCountry>
    <language>en</language>
    <displayLanguage>English</displayLanguage>
    <variant/>
    <displayVariant/>
  </supportedLocale>
```

```
  <supportedLocale>
    <fullyQualifiedName>fr_CA</fullyQualifiedName>
    <displayName>français (Canada)</displayName>
    <defaultLocale>false</defaultLocale>
    <country>CA</country>
    <displayCountry>Canada</displayCountry>
    <language>fr</language>
    <displayLanguage>français</displayLanguage>
    <variant/>
    <displayVariant/>
  </supportedLocale>
  <supportedLocale>
    <fullyQualifiedName>es_US</fullyQualifiedName>
    <displayName>español (Estados Unidos)</displayName>
    <defaultLocale>false</defaultLocale>
    <country>US</country>
    <displayCountry>Estados Unidos</displayCountry>
    <language>es</language>
    <displayLanguage>español</displayLanguage>
    <variant/>
    <displayVariant/>
  </supportedLocale>
</listSupportedLocalesResponse>
```

JSON response

```
{"listSupportedLocalesResponse":{"supportedLocale":
[{"defaultLocale":"true","displayCountry":"United States","displayVariant":
{},"displayLanguage":"English","fullyQualifiedName":"en_US","language":"en","displayName"
 (United States)","variant":{},"country":"US"},
{"defaultLocale":"false","displayCountry":"United Kingdom","displayVariant":
{},"displayLanguage":"English","fullyQualifiedName":"en_GB","language":"en","displayName"
 (United Kingdom)","variant":{},"country":"GB"},
{"defaultLocale":"false","displayCountry":"Canada","displayVariant":
{},"displayLanguage":"English","fullyQualifiedName":"en_CA","language":"en","displayName"
 (Canada)","variant":{},"country":"CA"},
{"defaultLocale":"false","displayCountry":"Canada","displayVariant":
{},"displayLanguage":"français","fullyQualifiedName":"fr_CA","language":"fr","displayName
 (Canada)","variant":{},"country":"CA"},
{"defaultLocale":"false","displayCountry":"Estados Unidos","displayVariant":
{},"displayLanguage":"español","fullyQualifiedName":"es_US","language":"es","displayName"
 (Estados Unidos)","variant":{},"country":"US"}]}}
```

Example

```
https://demo-secure.convio.net/demo829/site/CRContentAPI?
api_key=MyAPIKey&v=1.0&method=listSupportedLocales
```

DataSync Cross-Reference API

SRDataSyncAPI
`https://secure2.convio.net/organization/site/SRDataSyncAPI`

Description

The DataSync cross-reference API allows you to query and modify cross-reference metadata that maps objects within the Convio database to objects in another database.

The DataSync API exposes the following methods:

addXref

Add an XRef.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRDataSyncAPI	POST	Yes.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRDataSyncAPI?method=addXref &` `api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& convio_value =`*value* `& type_id =`*value* [`& description =`*value*] [`& foreign_value_0 =`*value*] [`& foreign_value_1 =`*value*] [`& foreign_value_2 =`*value*] [`& foreign_value_3 =`*value*] [`& foreign_value_4 =`*value*] [`& foreign_value_5 =`*value*] [`& foreign_value_6 =`*value*] [`& foreign_value_7 =`*value*] [`& foreign_value_8 =`*value*] [`& foreign_value_9 =`*value*]

Usage Notes

Adding an xref requires type_id, convio_value, and foreign values as required by the xref type. A description of the xref may optionally be provided. If successful the new xref is returned to the caller.

While the various foreign_value parameters are marked optional, they should be provided as required by the xref type definition. For example if an xref type has 3 fields, foreign_value_1, foreign_value_2, and foreign_value_3 should be defined. If a field is not required by the xref type definition, it may be omitted.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

convio_value

Required. COM object key

Type xsd:string.

type_id

Required. XRef type ID

Type xsd:int.

description

Optional. XRef description

Type xsd:string.

foreign_value_0

Optional. Foreign value

Type xsd:string.

foreign_value_1

Optional. Foreign value

Type xsd:string.

foreign_value_2

Optional. Foreign value

Type xsd:string.

foreign_value_3

Optional. Foreign value

Type xsd:string.

foreign_value_4

Optional. Foreign value

Type xsd:string.

foreign_value_5

Optional. Foreign value

Type xsd:string.

foreign_value_6

Optional. Foreign value

Type xsd:string.

foreign_value_7

Optional. Foreign value

Type xsd:string.

foreign_value_8

Optional. Foreign value

Type xsd:string.

foreign_value_9

Optional. Foreign value

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<addXrefResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <xref>
    <id>100</id>
    <xrefTypeId>400</xrefTypeId>
    <convioValue>com_object_key1</convioValue>
    <description>Sample XRef description</description>
    <securityCategoryId>1</securityCategoryId>
    <approvalStatus>submitted</approvalStatus>
    <createdBy>1010</createdBy>
    <createdOn>2013-02-03-06:00</createdOn>
    <lastChangeOn>2013-02-03-06:00</lastChangeOn>
    <foreignValue>
      <number>0</number>
      <value>foreign_key_1</value>
    </foreignValue>
    <foreignValue>
      <number>1</number>
      <value>foreign_key_2</value>
    </foreignValue>
  </xref>
</addXrefResponse>
```

JSON response


```
{"addXrefResponse":{"xref":
{"createdOn":"2013-02-03-06:00","id":"100","convioValue":"com_object_key1","foreignValue"
[{"value":"foreign_key_1","number":"0"},
{"value":"foreign_key_2","number":"1"}],"createdBy":"1010","xrefTypeId":"400","descriptio
 XRef
 description","approvalStatus":"submitted","lastChangeOn":"2013-02-03-06:00","securityCat
```


listComObjectsForXrefType

Given an XRef type ID, return a list of COM objects to which that XRef type can be applied.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRDataSyncAPI	SRDataSyncAPI	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRDataSyncAPI?method=listComObjectsForXrefType &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&type_id=`*value* [`&include_mapped=`*value*] [`&list_ascending=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRDataSyncAPI?method=listComObjectsForXrefType & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& type_id =`*value* [`& include_mapped =`*value*] [`& list_ascending =`*value*] [`& list_filter_text =`*value*] [`& list_page_offset =`*value*] [`& list_page_size =`*value*] [`& list_record_offset =`*value*] [`& list_sort_column =`*value*]

Usage Notes

Given a type_id, returns a list of Convio objects to which that xref type can be applied. Objects are formatted with display names and internal values, which should be used when creating an xref with the *addXref* method.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

type_id

Required. XRef type ID

Type xsd:int.

include_mapped

Optional. Include Mapped

Type xsd:boolean.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

type_id

Required. XRef type ID

Type xsd:int.

include_mapped

Optional. Include Mapped

Type xsd:boolean.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listComObjectsForXrefTypeResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>2</totalNumberResults>
  <comObject>
    <name>COM Object 1</name>
    <value>100</value>
  </comObject>
  <comObject>
    <name>COM Object 2</name>
    <value>101</value>
  </comObject>
</listComObjectsForXrefTypeResponse>
```

JSON response

```
{"listComObjectsForXrefTypeResponse":{"totalNumberResults":"2","comObject":
[{"name":"COM Object 1","value":"100"},{"name":"COM Object
 2","value":"101"}]}}
```

listForeignObjectsForXrefField

Get name/value pairs of foreign objects valid for a field in an XRef type.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRDataSyncAPI	SRDataSyncAPI	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRDataSyncAPI?method=listForeignObjectsForXrefField &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*

][&response_format=*xml* | *json*][&sign_redirects=*value*][&source=*value*][&sub_source=*value*][&success_redirect=*value*][&suppress_response_codes=*value*] &field_number=*value* &type_id=*value* [&list_ascending=*value*][&list_filter_text=*value*][&list_page_offset=*value*][&list_page_size=*value*][&list_record_offset=*value*][&list_sort_column=*value*][&auth=*value*][&sso_auth_token=*value*]

Server API Syntax

https://*secure2.convio.net/organization*/site/SRDataSyncAPI?method=listForeignObjectsForXrefField & api_key =*value* & login_name =*value* & login_password =*value* & v =*value* [& center_id =*value*][& response_format =*xml* |*json*][& source =*value*][& sub_source =*value*][& suppress_response_codes =*value*] & field_number =*value* & type_id =*value* [& list_ascending =*value*][& list_filter_text =*value*][& list_page_offset =*value*][& list_page_size =*value*][& list_record_offset =*value*][& list_sort_column =*value*]

Usage Notes

Given a type_id and field_number, returns a list of foreign objects which apply to that xref type and field. Objects are formatted with display names and internal values, which should be used when creating an xref with the *addXref* method.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

field_number

Required. Field Number

Type xsd:int.

type_id

Required. XRef type ID

Type xsd:int.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

field_number

Required. Field Number

Type xsd:int.

type_id

Required. XRef type ID

Type xsd:int.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<listForeignObjectsForXrefFieldResponse xsi:schemaLocation="http://convio.com/
crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>2</totalNumberResults>
  <foreignObject>
    <name>Foreign Object 1</name>
    <value>foreign_value_1</value>
  </foreignObject>
  <foreignObject>
    <name>Foreign Object 2</name>
    <value>foreign_value_2</value>
  </foreignObject>
</listForeignObjectsForXrefFieldResponse>
```

JSON response

```
{"listForeignObjectsForXrefFieldResponse":{"foreignObject":[{"name":"Foreign
 Object 1","value":"foreign_value_1"},{"name":"Foreign Object
 2","value":"foreign_value_2"}],"totalNumberResults":"2"}}
```

listXrefTypes

Get XREF types for a COM application.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRDataSyncAPI	GET, POST	Yes.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRDataSyncAPI?method=listXrefTypes &` `api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] [`& application =`*donation_classic* | *donation_campaign* | *donation_form* | *teamraiser_campaign* | *ticketed_event* | *pf_campaign* | *action_alert* | *email_campaign* | *designated_gift* | *e_commerce_store* | *event_ticket_class* | *event_rsvp* | *interest* | *membership_type*]

Usage Notes

Given an application, this method returns the xref types assigned to that application.

Given an xref type ID, a list of name/value pairs of Convio objects to which that xref type may apply can be generated using the *listComObjectsForXrefType* method.

The same can be done for foreign objects given xref type id and field number using the *listForeignObjectsForXrefField* method.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

application

Optional. COM application

Type crm:comApplication.

Options are:

- donation_classic
- donation_campaign
- donation_form
- teamraiser_campaign

- ticketed_event
- pf_campaign
- action_alert
- email_campaign
- designated_gift
- e_commerce_store
- event_ticket_class
- event_rsvp
- interest
- membership_type

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listXrefTypesResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <xrefType>
    <id>100</id>
    <name>Sample XRef Type</name>
    <description>A sample xref type.</description>
    <xrefField>
      <number>0</number>
      <name>Sample field name</name>
      <required>false</required>
    </xrefField>
  </xrefType>
</listXrefTypesResponse>
```

JSON response

```
{"listXrefTypesResponse":{"xrefType":{"id":"100","xrefField":{"name":"Sample
 field name","number":"0","required":"false"},"description":"A sample xref
 type.","name":"Sample XRef Type"}}}
```

listXrefs

Get XRefs given search criteria.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRDataSyncAPI	GET, POST	Yes.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRDataSyncAPI?method=listXrefs &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] [`& convio_value =`*value*] [`& list_page_offset =`*value*] [`& list_page_size =`*value*] [`& list_record_offset =`*value*] [`& type_id =`*value*] [`& xref_id =`*value*]

Usage Notes

Retrieves a list of xrefs, optionally filtered by xref_id, type_id, or convio_value.

This method may also be called without parameters to retrieve all xrefs. When providing xref_id, type_id and convio_value will be ignored.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

convio_value

Optional. COM object key

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

type_id

Optional. XRef type ID

Type xsd:int.

xref_id

Optional. XRef ID

Type xsd:int.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listXrefsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <xref>
    <id>100</id>
    <xrefTypeId>400</xrefTypeId>
    <convioValue>com_object_key1</convioValue>
    <description>Sample XRef description</description>
    <securityCategoryId>1</securityCategoryId>
    <approvalStatus>submitted</approvalStatus>
    <createdBy>1010</createdBy>
    <createdOn>2013-02-03-06:00</createdOn>
    <lastChangeOn>2013-02-03-06:00</lastChangeOn>
    <foreignValue>
      <number>0</number>
      <value>foreign_key_1</value>
    </foreignValue>
    <foreignValue>
```

```
      <number>1</number>
      <value>foreign_key_2</value>
    </foreignValue>
  </xref>
  <xref>
    <id>101</id>
    <xrefTypeId>402</xrefTypeId>
    <convioValue>com_object_key2</convioValue>
    <description>Sample XRef description</description>
    <securityCategoryId>1</securityCategoryId>
    <approvalStatus>approved</approvalStatus>
    <createdBy>1011</createdBy>
    <createdOn>2013-02-03-06:00</createdOn>
    <lastChangeOn>2013-02-03-06:00</lastChangeOn>
    <foreignValue>
      <number>0</number>
      <value>foreign_key_3</value>
    </foreignValue>
    <foreignValue>
      <number>1</number>
      <value>foreign_key_4</value>
    </foreignValue>
  </xref>
</listXrefsResponse>
```

JSON response

```
{"listXrefsResponse":{"xref":
[{"createdOn":"2013-02-03-06:00","id":"100","convioValue":"com_object_key1","foreignValue
[{"value":"foreign_key_1","number":"0"},
{"value":"foreign_key_2","number":"1"}],"createdBy":"1010","xrefTypeId":"400","descriptio
 XRef
 description","approvalStatus":"submitted","lastChangeOn":"2013-02-03-06:00","securityCat
{"createdOn":"2013-02-03-06:00","id":"101","convioValue":"com_object_key2","foreignValue"
[{"value":"foreign_key_3","number":"0"},
{"value":"foreign_key_4","number":"1"}],"createdBy":"1011","xrefTypeId":"402","descriptio
 XRef
 description","approvalStatus":"approved","lastChangeOn":"2013-02-03-06:00","securityCate
```

removeXref

Remove an XRef.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRDataSyncAPI	POST	Yes.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRDataSyncAPI?method=removeXref &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& xref_id =`*value*

Usage Notes

Removes an xref given xref_id. The removed xref is returned to the caller.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

xref_id

Required. XRef ID

Type xsd:int.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<removeXrefResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <xref>
    <id>100</id>
    <xrefTypeId>400</xrefTypeId>
    <convioValue>com_object_key1</convioValue>
    <description>Sample XRef description</description>
    <securityCategoryId>1</securityCategoryId>
    <approvalStatus>submitted</approvalStatus>
    <createdBy>1010</createdBy>
    <createdOn>2013-02-03-06:00</createdOn>
    <lastChangeOn>2013-02-03-06:00</lastChangeOn>
    <foreignValue>
      <number>0</number>
      <value>foreign_key_1</value>
    </foreignValue>
    <foreignValue>
      <number>1</number>
      <value>foreign_key_2</value>
    </foreignValue>
  </xref>
</removeXrefResponse>
```

JSON response

```
{"removeXrefResponse":{"xref":
{"createdOn":"2013-02-03-06:00","id":"100","convioValue":"com_object_key1","foreignValue"
[{"value":"foreign_key_1","number":"0"},
{"value":"foreign_key_2","number":"1"}],"createdBy":"1010","xrefTypeId":"400","descriptio
 XRef
 description","approvalStatus":"submitted","lastChangeOn":"2013-02-03-06:00","securityCat
```

Donation API

CRDonationAPI
`https://secure2.convio.net/organization/site/CRDonationAPI`

Description

The Donation API allows processing of credit card online donations online, and logging of donations processed offline. The donation API methods operate on "Donations 2" (not "Donations Classic") shadow donation forms.

The `donate` method, used to process online credit card donations, supports only Client API calls. To maintain PCI data security standards for credit card transactions, you must call this method directly from the client's browser and never proxy the call through a non-Convio server.

The `addOfflineDonation` and `refundOfflineDonation` methods record monetary transactions processed outside the Convio system. These methods provide only Server APIs and are accessible only to white-listed servers. You must create an API Administrator account to call these methods and assign it to an Administrative Security Group having appropriate permissions of the "Offline Donation API" permission type. The built-in "API Administrators" group does not have "Offline Donation API" permissions by default. If these APIs will be used by a third-party payment processor to record transactions on your site, you should limit the permissions on the accounts you create for that purpose, and not grant additional permissions by assigning these accounts to the "API Administrators" group.

addOfflineDonation

This method processes an offline donation by a constituent to a Donations 2 (not "Donations Classic") offline/online or offline only donation form.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRDonationAPI	POST	No.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRDonationAPI?method=addOfflineDonation` & `api_key` =*value* & `login_name` =*value* & `login_password` =*value* & `v` =*value* [& `center_id` =*value*] [& `response_format` =*xml* |*json*] [& `source` =*value*] [& `sub_source` =*value*] [& `suppress_response_codes` =*value*] & `billing.name.first` =*value* & `billing.name.last` =*value* & `donor.email` =*value* & `form_id` =*value* & `level_id` =*value* & `offline_payment_method` =*cash* |*check* [& `additional_amount` =*value*] [& `anonymous` =*value*] [& `billing.address.city` =*value*] [& `billing.address.country` =*value*] [& `billing.address.county` =*value*] [& `billing.address.state` =*value*] [& `billing.address.street1` =*value*] [& `billing.address.street2` =*value*] [& `billing.address.street3` =*value*] [& `billing.address.zip` =*value*] [& `billing.name.middle` =*value*] [& `billing.name.profSuffix` =*value*] [& `billing.name.suffix` =*value*] [& `billing.name.title` =*value*] [& `check_number` =*value*] [& `designated.X.amount` =*value*] [& `designated.X.id` =*value*] [& `designated_write_in.X.amount` =*value*] [& `designated_write_in.X.contact` =*value*] [& `designated_write_in.X.name` =*value*] [& `donor.address.city` =*value*] [& `donor.address.country` =*value*] [& `donor.address.county` =*value*] [& `donor.address.state` =*value*] [& `donor.address.street1` =*value*] [& `donor.address.street2` =*value*] [& `donor.address.street3` =*value*] [& `donor.address.zip` =*value*] [& `donor.email_opt_in` =*value*] [& `donor.employer` =*value*] [& `donor.name.first` =*value*] [& `donor.name.last` =*value*] [& `donor.name.middle` =*value*] [& `donor.name.profSuffix` =*value*] [& `donor.name.suffix` =*value*] [& `donor.name.title` =*value*] [& `donor.occupation` =*value*] [& `donor.phone` =*value*] [& `donor.phone_type` =*home* |*work* |*other*] [& `gift_aid` =*value*] [& `joint_donor.name.first` =*value*] [& `joint_donor.name.last` =*value*] [& `joint_donor.name.middle` =*value*] [& `joint_donor.name.suffix` =*value*] [& `joint_donor.name.title` =*value*] [& `matching_eligible` =*value*] [& `other_amount` =*value*] [& `premium_id` =*value*] [& `send_receipt` =*value*] [& `send_registration_email` =*value*] [& `shipping.address.city` =*value*] [& `shipping.address.country` =*value*] [& `shipping.address.county` =*value*] [& `shipping.address.state` =*value*] [& `shipping.address.street1` =*value*] [& `shipping.address.street2` =*value*] [& `shipping.address.street3` =*value*] [& `shipping.address.zip` =*value*] [& `shipping.email` =*value*] [& `shipping.name.first` =*value*] [& `shipping.name.last` =*value*] [& `shipping.name.middle` =*value*] [& `shipping.name.profSuffix` =*value*] [& `shipping.name.suffix` =*value*] [& `shipping.name.title` =*value*] [& `shipping.phone` =*value*] [& `validate` =*value*]

Usage Notes

The `addOfflineDonation` method allows offline donations (meaning transactions that are processed by an external payment processor) to be recorded in Convio.

The *refundOfflineDonation* method may be used to refund transactions logged through this API.

In a typical use case, a third-party payment service provider might use this API to notify Convio of donation and refund transactions on behalf of a client. Because the same third-third party service provider may implement this API for multiple clients, clients should strictly limit the permissions of the API Administrator account that has access to this API.

API Administrator accounts that access this API must be assigned the Offline Donation API permission type, under which they may be granted permission either to "Add Offline Donations via API" or to "Add and Refund Offline Donations via API". You should grant these accounts only the minimum required permissions.

Note: API Administrator accounts are limited to API access, they are never permitted to log in to Convio.

All donation methods must reference a shadow Donation Form on the Convio web site. This shadow form is used to perform validation and to associate donations with the appropriate fundraising Campaign. The ID of this shadow Donation Form is passed in the required form_id parameter. The Donation Form must be designated as either online/offline or offline-only. The API also requires the ID of the appropriate Donation Level in the level_id parameter.

Validation can either simply check for the minimally required fields necessary to process a transaction (validate=*false*) or it can perform all of the same field-level validation done by the a Convio donation form when accessed via its web page (validate=*true*). In addition to the parameters marked as always being required, the following conditions must also be met:

- There must be a valid donor_email value.

Designated Giving

The Designated Giving feature in Online Giving allows you to set up Designees within your organization and then assign all funds from a donation form to a specific designee or permit your donors to specify which Designee(s) they would like to target with their gifts. To use this feature, you must first set up your list of Designees and Designation Types. Designation Types help organize your designees for display to your donors. For example, an educational institution might create the following Designation Types: Academic, Athletic, and Administrative. Individual designees, such as the football program or the biology department, can then be assigned to the appropriate Type.

To get the designation types and designees to use for a form, call the *getDesignationTypes* and *getDesignees* methods. To specify designations for a donation, use the designated.X.id and designated.X.amount parameters, substituting numbers for the "X" part of the parameter names to identify ID and amount pairs (the numbers do not need to be sequential). To specify "write-in" designations, use the designated_write_in.X.name, designated_write_in.X.contact, and designated_write_in.X.amount parameters.

Donation Errors

If the call to the API succeeds but one or more errors are encountered during transaction processing, a `donationResponse` object will be returned listing reasons for the error. To correlate with the standard donation form processing, `pageError` elements in the response are the same messages that would be displayed near the top of the page and are sometimes generic. The `fieldError` elements in the response are the same messages that would be displayed near input fields throughout the page and are always specific to the validation of input for an input field.

Error Responses

For common API errors such as authentication errors, the common error response is returned. An XML example:

```
<errorResponse xmlns="http://convio.com/crm/v1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://convio.com/crm/v1.0 http://service.convio.net/
xmlschema/crm.public.v1.xsd">
  <code>2</code>
  <message>Incorrect API key.  Verify that the value of the parameter api_key
 matches the value of the SDP CONVIO_API_KEY.</message>
</errorResponse>
```

A JSON example:

```
{"errorResponse":{
  "code": "2",
  "message": "Incorrect API key.  Verify that the value of the parameter
 api_key matches the value of the SDP CONVIO_API_KEY.",
}}
```

For donation-specific errors, the "donationResponse" is returned and contains a list of the error messages. An XML example:

```
<donationResponse xmlns="http://convio.com/crm/v1"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://convio.com/crm/v1.0 http://service.convio.net/
xmlschema/crm.public.v1.xsd">
  <errors>
    <code>101</code>
    <message>Error: There was a problem encountered while processing your
 donation.</message>
    <reason>FIELD_VALIDATION</reason>
    <pageError>There was a problem processing your request.  Please see
 below.</pageError>
    <fieldError>An email address is required.</fieldError>
    <fieldError>Billing state or province is required.</fieldError>
    <fieldError>Billing last name is required.</fieldError>
    <fieldError>Billing zip or postal code is required.</fieldError>
    <fieldError>Billing street address is required.</fieldError>
    <fieldError>Billing first name is required.</fieldError>
    <fieldError>Billing city is required.</fieldError>
  </errors>
</donationResponse>
```

A JSON example:

```
{"donationResponse":
  {"errors":
    {"code":"101",
     "reason":"FIELD_VALIDATION",
     "message":"Error: There was a problem encountered while processing your
 donation.",
     "fieldError":["An email address is required.",
                   "Billing state or province is required.",
                   "Billing last name is required.",
                   "Billing zip or postal code is required.",
                   "Billing street address is required.",
                   "Billing first name is required.",
                   "Billing city is required."],
     "pageError":"There was a problem processing your request.  Please see
 below."}
  }
}
```

Note: If you specify "true" in the optional suppress_response_codes parameter, both success and error results will return a donationResponse document

with an HTTP return code of 200 - OK. You must parse the donationResponse document for an errors element to determine whether or not the call succeeded.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

billing.name.first

Required. The billing name of the credit card owner. If not supplied, defaults to donor.name.first.

Type xsd:string.

billing.name.last

Required. The billing name of the credit card owner. If not supplied, defaults to donor.name.last.

Type xsd:string.

donor.email

Required. The donor's email address.

Type xsd:string.

form_id

Required. The ID of the donation form.

Type xsd:nonNegativeInteger.

level_id

Required. The ID of the donation level.

Type xsd:nonNegativeInteger.

offline_payment_method

Required. Payment method for this offline transaction

Type xsd:string.

Default is cash.

Options are:

- cash
- check

additional_amount

Optional. Donation amount in addition to level amount.

Type xsd:string.

anonymous

Optional. Whether to mark the donation as "anonymous".

Type xsd:boolean.

billing.address.city

Optional. The billing address of the credit card owner. If not supplied, defaults to donor.address.city.

Type xsd:string.

billing.address.country

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.address.county

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.address.state

Optional. The billing address of the credit card owner. If not supplied, defaults to donor.address.state.

Type xsd:string.

billing.address.street1

Optional. The billing address of the credit card owner. If not supplied, defaults to donor.address.street1.

Type xsd:string.

billing.address.street2

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.address.street3

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.address.zip

Optional. The billing address of the credit card owner. If not supplied, defaults to donor.address.zip.

Type xsd:string.

billing.name.middle

Optional. The billing name of the credit card owner.

Type xsd:string.

billing.name.profSuffix

Optional. The billing name of the credit card owner.

Type xsd:string.

billing.name.suffix

Optional. The billing name of the credit card owner.

Type xsd:string.

billing.name.title

Optional. The billing name of the credit card owner.

Type xsd:string.

check_number

Optional. Check number for an offline donation made by check

Type xsd:string.

designated.X.amount

Optional. The amount of a donation to designate to a designee. The "X" part of the parameter must be an integer used to correspond to designated.X.id or, in the case of write-in designations,

designated.X.name and designated.X.contact. This allows multiple pairs of ID/amount for multiple designations.

Type xsd:nonNegativeInteger.

designated.X.id

Optional. The ID of a designee to which to designate all or part of a donation. The "X" part of the parameter must be an integer used to correspond to designated.X.amount. This allows multiple pairs of ID/amount for multiple designations.

Type xsd:nonNegativeInteger.

designated_write_in.X.amount

Optional. The amount of a donation to designate to a designee. The "X" part of the parameter must be an integer used to correspond to designated_write_in.X.name and designated_write_in.X.contact. This allows multiple pairs of ID/amount for multiple designations.

Type xsd:nonNegativeInteger.

designated_write_in.X.contact

Optional. The contact information of an organization to which to designate all or part of a donation. The "X" part of the parameter must be an integer used to correspond to designated_write_in.X.name and designated_write_in.X.amount. This allows multiple sets of name/contact/amount for multiple designations.

Type xsd:string.

designated_write_in.X.name

Optional. The name of an organization to which to designate all or part of a donation. The "X" part of the parameter must be an integer used to correspond to designated_write_in.X.contact and designated_write_in.X.amount. This allows multiple sets of name/contact/amount for multiple designations.

Type xsd:string.

donor.address.city

Optional. The donor's address.

Type xsd:string.

donor.address.country

Optional. The donor's address.

Type xsd:string.

donor.address.county

Optional. The donor's address.

Type xsd:string.

donor.address.state

Optional. The donor's address.

Type xsd:string.

donor.address.street1

Optional. The donor's address.

Type xsd:string.

donor.address.street2

Optional. The donor's address.

Type xsd:string.

donor.address.street3

Optional. The donor's address.

Type xsd:string.

donor.address.zip

Optional. The donor's address.

Type xsd:string.

donor.email_opt_in

Optional. The donor's email opt-in selection.

Type xsd:string.

donor.employer

Optional. The donor's employer.

Type xsd:string.

donor.name.first

Optional. The donor's name.

Type xsd:string.

donor.name.last

Optional. The donor's name.

Type xsd:string.

donor.name.middle

Optional. The donor's name.

Type xsd:string.

donor.name.profSuffix

Optional. The donor's name.

Type xsd:string.

donor.name.suffix

Optional. The donor's name.

Type xsd:string.

donor.name.title

Optional. The donor's name.

Type xsd:string.

donor.occupation

Optional. The donor's occupation.

Type xsd:string.

donor.phone

Optional. The donor's phone number.

Type xsd:string.

donor.phone_type

Optional. The type of the donor's phone number .

Type xsd:string.

Options are:

- home
- work
- other

gift_aid

Optional. Gift aid status (UK Only).

Type xsd:string.

joint_donor.name.first

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.last

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.middle

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.suffix

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.title

Optional. The name for a joint donor.

Type xsd:string.

matching_eligible

Optional. Whether or not the donation is eligible for a matching gift.

Type xsd:boolean.

other_amount

Optional. User specified donation amount.

Type xsd:string.

premium_id

Optional. Unique ID of premium.

Type xsd:nonNegativeInteger.

send_receipt

Optional. Whether to send a receipt email to the donor. The default behavior is to send a receipt.

Type xsd:boolean.

Default is true.

send_registration_email

Optional. If a new constituent record is created for the donor, this specifies whether to send a registration "welcome" email to the donor. The default behavior is to send it.

Type xsd:boolean.

Default is true.

shipping.address.city

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.country

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.county

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.state

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.street1

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.street2

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.street3

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.zip

Optional. Shipping address for premium.

Type xsd:string.

shipping.email

Optional. Email address of the recipient.

Type xsd:string.

shipping.name.first

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.last

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.middle

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.profSuffix

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.suffix

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.title

Optional. Premium recipient's name.

Type xsd:string.

shipping.phone

Optional. Phone number of the recipient.

Type xsd:string.

validate

Optional. Whether to enforce validation of every field according to the specification in the ghost form ("true" or "false"). Some validation is always in effect, such as ensuring that the campaign, form, and level IDs reference published entities and that there is sufficient information to process the transaction.

Type xsd:boolean.

Default is false.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<donationResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <donation>
    <cons_id>1006789</cons_id>
    <confirmation_code>0-0000-0-0000-0000</confirmation_code>
    <date_time>2013-02-03T05:18:27.883-06:00</date_time>
    <org_tax_id>00-0000000</org_tax_id>
    <reward_points_earned>100</reward_points_earned>
    <transaction_id>100101</transaction_id>
    <amount>
      <decimal>100</decimal>
      <formatted>$100.00</formatted>
    </amount>
    <tax_deductible_amount>
      <decimal>100</decimal>
      <formatted>$100.00</formatted>
    </tax_deductible_amount>
    <value_of_goods>
      <decimal>0</decimal>
      <formatted>$0.00</formatted>
    </value_of_goods>
    <summary_data>
```

```
      <field header="false">
        <label>Transaction Date:</label>
        <value>0/0/201</value>
        <name>date_id</name>
      </field>
      <field header="true">
        <label>Gift Information</label>
        <name>section-header</name>
      </field>
      <field header="false">
        <label>Amount:</label>
        <value>$100.00</value>
        <name>level_id</name>
      </field>
      <field header="true">
        <label>Billing Information</label>
        <name>section-header</name>
      </field>
      <field header="false">
        <label>Billing First Name:</label>
        <value>John</value>
        <name>billing.name.first</name>
      </field>
      <field header="false">
        <label>Billing Last Name:</label>
        <value>Doe</value>
        <name>billing.name.last</name>
      </field>
    </summary_data>
  </donation>
</donationResponse>
```

JSON response


```
{"donationResponse":{"donation":{"amount":
{"formatted":"$100.00","decimal":"100"},"value_of_goods":
{"formatted":"$0.00","decimal":"0"},"summary_data":{"field":
[{"name":"date_id","value":"0/0/201","label":"Transaction
 Date:","header":"false"},{"name":"section-header","label":"Gift
 Information","header":"true"},
{"name":"level_id","value":"$100.00","label":"Amount:","header":"false"},
{"name":"section-header","label":"Billing Information","header":"true"},
{"name":"billing.name.first","value":"John","label":"Billing First
 Name:","header":"false"},
{"name":"billing.name.last","value":"Doe","label":"Billing
 Last Name:","header":"false"}]},"tax_deductible_amount":
{"formatted":"$100.00","decimal":"100"},"org_tax_id":"00-0000000","confirmation_code":"0
```


calculateNextPaymentDate

Given a date, calcualates the next payment for a sustainer gift.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRRecurringAPI	SRRecurringAPI	POST	Yes.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRRecurringAPI?method=calculateNextPaymentDate &recurring_gift_id=`*value* `&next_payment_date=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]
Server API Syntax
`https://`*secure2.convio.net/organization*`/site/SRRecurringAPI?method=calculateNextPaymentDate & recurring_gift_id =`*value* `& next_payment_date =`*value* `& cons_id =`*value*

Usage Notes

This method calculates the next payment date in the series for a recurring gift.

You may use the following methods to allow a logged-in user to manage their recurring gifts themselves via the Client API or to manage recurring gifts as an adminstrator via the Server API:

- *calculateNextPaymentDate*
- *cancelRecurringGift*
- *getRecurringConfiguration*
- *getRecurringGiftDetails*
- *getRecurringGifts*
- *getRecurringHistory*
- *skipNextPayment*
- *updateRecurringBankAccountInfo*
- *updateRecurringCreditCardInfo*
- *updateRecurringGift_method*

Client API Parameters
Common Client Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

next_payment_date

Required. Specifies the next payment date of a recurring gift. The date must be in the format MM-DD-YYYY

Type xsd:nonNegativeInteger.

Server API Parameters
Common Server Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

next_payment_date

Required. Specifies the next payment date of a recurring gift. The date must be in the format MM-DD-YYYY

Type xsd:nonNegativeInteger.

cons_id

Required. Specifies the ID of the constiuent.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<calculateNextPaymentDateResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
</calculateNextPaymentDateResponse>
```

JSON response

```
{"calculateNextPaymentDateResponse":{"success":"true"}}
```

cancelRecurringGift

Cancels a recurring gift.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRRecurringAPI	SRRecurringAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRRecurringAPI?method=cancelRecurringGift &recurring_gift_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRRecurringAPI?method=cancelRecurringGift & recurring_gift_id =`*value* `& cons_id =`*value*

Usage Notes

This method allows a logged-in user or an administrative process to cancel all future payments on a recurring gift or pledge. It does not refund payments already processed. To refund payments use *refundTransaction*.

You may use the following methods to allow a logged-in user to manage their recurring gifts themselves via the Client API or to manage recurring gifts as an adminstrator via the Server API:

- *calculateNextPaymentDate*
- *cancelRecurringGift*
- *getRecurringConfiguration*
- *getRecurringGiftDetails*
- *getRecurringGifts*
- *getRecurringHistory*
- *skipNextPayment*
- *updateRecurringBankAccountInfo*
- *updateRecurringCreditCardInfo*
- *updateRecurringGift_method*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

cons_id

Required. Specifies the ID of the constiuent.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<cancelRecurringGiftResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <canceled>true</canceled>
</cancelRecurringGiftResponse>
```

JSON response

```
{"cancelRecurringGiftResponse":{"canceled":"true"}}
```

donate

This method processes a donation by a constituent to a Donations 2 (not "Donations Classic") donation form.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRDonationAPI	None.	POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRDonationAPI?method=donate`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&billing.address.city=`*value* `&billing.address.state=`*value* `&billing.address.street1=`*value* `&billing.address.zip=`*value* `&billing.name.first=`*value* `&billing.name.last=`*value* `&donor.email=`*value* `&form_id=`*value* `&level_id=`*value* [`&ach_account=`*value*] [`&ach_account_type=`*CHECKING* | *SAVINGS*] [`&ach_bank=`*value*] [`&ach_routing=`*value*] [`&ach_transit=`*value*] [`&additional_amount=`*value*] [`&anonymous=`*value*] [`&billing.address.country=`*value*] [`&billing.address.county=`*value*] [`&billing.address.street2=`*value*] [`&billing.address.street3=`*value*] [`&billing.name.middle=`*value*] [`&billing.name.profSuffix=`*value*] [`&billing.name.suffix=`*value*] [`&billing.name.title=`*value*] [`&card_cvv=`*value*] [`&card_exp_date=`*value*] [`&card_exp_date_month=`*value*] [`&card_exp_date_year=`*value*] [`&card_number=`*value*] [`&designated.X.amount=`*value*] [`&designated.X.id=`*value*] [`&designated_write_in.X.amount=`*value*] [`&designated_write_in.X.contact=`*value*

][&designated_write_in.X.name=*value*][&donor.address.city=*value*][&donor.address.country=*value*][&donor.address.county=*value*][&donor.address.state=*value*][&donor.address.street1=*value*][&donor.address.street2=*value*][&donor.address.street3=*value*][&donor.address.zip=*value*][&donor.email_opt_in=*value*][&donor.employer=*value*][&donor.name.first=*value*][&donor.name.last=*value*][&donor.name.middle=*value*][&donor.name.profSuffix=*value*][&donor.name.suffix=*value*][&donor.name.title=*value*][&donor.occupation=*value*][&donor.phone=*value*][&donor.phone_type=*home* | *work* | *other*][&ecard.copy_sender=*value*][&ecard.id=*value*][&ecard.message=*value*][&ecard.recipients=*value*][&ecard.send=*value*][&ecard.send_date=*value*][&ecard.subject=*value*][&fr_id=*value*][&gift_aid=*value*][&installment.duration=*value*][&installment.frequency=*one-time* | *monthly* | *quarterly* | *semi-annually* | *annually*][&joint_donor.name.first=*value*][&joint_donor.name.last=*value*][&joint_donor.name.middle=*value*][&joint_donor.name.suffix=*value*][&joint_donor.name.title=*value*][&level_autorepeat=*value*][&matching_eligible=*value*][&other_amount=*value*][&premium_id=*value*][&premium_option=*value*][&remember_me=*value*][&send_receipt=*value*][&send_registration_email=*value*][&shipping.address.city=*value*][&shipping.address.country=*value*][&shipping.address.county=*value*][&shipping.address.state=*value*][&shipping.address.street1=*value*][&shipping.address.street2=*value*][&shipping.address.street3=*value*][&shipping.address.zip=*value*][&shipping.email=*value*][&shipping.name.first=*value*][&shipping.name.last=*value*][&shipping.name.middle=*value*][&shipping.name.profSuffix=*value*][&shipping.name.suffix=*value*][&shipping.name.title=*value*][&shipping.phone=*value*][&soft_credit_id=*value*][&soft_credit_type=*TR_PARTICIPANT* | *TR_TEAM* | *TR_EVENT* | *TRIBUTE_GIFT*][&summary=*data* | *page* | *both*][&sustaining.duration=*value*][&sustaining.frequency=*one-time* | *monthly* | *quarterly* | *semi-annually* | *annually*][&teamraiser.message_to_participant=*value*][&teamraiser.recognition_name=*value*][&teamraiser.show_gift_to_public=*value*][&tribute.honoree.deceased=*value*][&tribute.honoree.name.first=*value*][&tribute.honoree.name.full=*value*][&tribute.honoree.name.last=*value*][&tribute.honoree.name.title=*value*][&tribute.message.body=*value*][&tribute.message.closing=*value*][&tribute.message.include_amount=*value*][&tribute.message.signature=*value*][&tribute.notify.address.city=*value*][&tribute.notify.address.country=*value*][&tribute.notify.address.county=*value*][&tribute.notify.address.state=*value*][&tribute.notify.address.street1=*value*][&tribute.notify.address.street2=*value*][&tribute.notify.address.street3=*value*][&tribute.notify.address.zip=*value*][&tribute.notify.name.full=*value*][&tribute.notify.name.title=*value*][&tribute.type=*memorial* | *tribute*][&validate=*value*]

Usage Notes

Important: This method is used to process online credit card donations. It supports only Client API calls. To maintain PCI data security standards for payment card transactions, you must call this method directly from the client's browser and **never** proxy the call through a non-Convio server.

The Donation API must reference a shadow Donation Form on the Convio web site. This shadow form is used to perform validation and to associate donations with the appropriate fundraising Campaign. The ID of this shadow Donation Form is passed in the required form_id parameter. The API also requires the ID of the appropriate Donation Level in the level_id parameter.

The `donate` method is intended specifically to enable online donation processing by payment card or bank transfer. To process payments via other online payment services (PayPal or Amazon), you must use instead *startDonation*. These methods do not support eCommerce purchases, TeamRaiser registration fees, additional registration gifts or offline gifts, or Personal Fundraising gifts.

Other API methods may also process payment card transactions, including:

- *addGift*
- *processRegistration*

- *refundTransaction*
- *startDonation*

Effect on Session State and Constituent Record

If `donate` is called in the context of a logged-in user session, the transaction will be credited to the logged-in constituent and the donor's constituent record will be updated with the name and address information provided in the call (billing information fields are used if the donor information fields are not included in the form).

If `donate` is called in the context of a session established by a remember-me cookie, the donation is credited to the constituent identified by the session, and updates are made to some tracking and status fields, however the constituent record is not updated with new contact information. The same is true when `donate` is called outside of any session context, but when the donor's email address matches exactly one existing constituent record.

Otherwise, if `donate` is called without a user session and the donor's email address cannot be matched to an existing constituent, a new constituent record will be created for the donor, a session will be established for the new constituent, and a session cookie will be pushed to the caller.

Validation

Validation can either simply check for the minimally required fields necessary to process a transaction (validate=*false*) or it can perform all of the same field-level validation done by the a Convio donation form when accessed via its web page (validate=*true*). In addition to the parameters marked as always being required, the following conditions must also be met:

- There must be either credit card or ACH payment options specified.
- There must be a billing name and address for transaction verification. These may be passed either as billing or donor parameters. If billing parameters (billing.name.first, billing.name.last, billing.address.street1, billing.address.city, billing.address.state, and billing.address.zip) are specified, those take precedence, otherwise donor parameter values are used.

  The reason for two sets of name and address fields is to allow the constituent record of the donor to be different from the name and address used for payment processing.
- There must be a valid donor_email value.

Important: For security reasons, payment information may not be passed in the POST URL, and must be passed as multi-part form data of type `Content-type: application/x-www-form-urlencoded` in the request body.

Note: For testing purposes, you may use card_number=*4111111111111111*, card_cvv=*111*, and any date later than the transaction date for card_exp_date or card_exp_date_month and card_exp_date_year. Pass the df_preview parameter with any value when testing (the parameter value is not checked, only the presence of the parameter).

Note: Depending upon how the underlying donation form is configured, the tax-exempt amount returned may be different from the total amount of the donation. For example, if the donation is associated with a premium, the fair market value of the premium may be deducted.

Boolean Parameters in HTML Forms

Boolean parameters such as donor.email_opt_in and remember_me expect a value of "true" (ignoring case) to be set. Any other value is interpreted as "false". To set this in an HTML checkbox, include `value="true"` in the checkbox attributes.

Custom Fields

In the Donation Management application, administrators can define new custom fields to add to donation forms. To set a custom field with this method, the name of the API parameter is derived from the "Data Element Name" provided for the custom field. The Data Element Name is converted into a valid identifier by converting all letters to lower case and replacing all non-alphanumeric characters with underscores. So a Data Element Name of "My Custom

String" should be passed into the donation API as "my_custom_string". Note that the "Data Element Name" is the same across all donation forms, and is not the same as the label that you actually specify in the form.

If validation is disabled (validate=*false*) and a Custom Field contains text longer than the maximum length for the field the text will be truncated to the maximum length and saved without an error.

Designated Giving

The Designated Giving feature in Online Giving allows you to set up Designees within your organization and then assign all funds from a donation form to a specific designee or permit your donors to specify which Designee(s) they would like to target with their gifts. To use this feature, you must first set up your list of Designees and Designation Types. Designation Types help organize your designees for display to your donors. For example, an educational institution might create the following Designation Types: Academic, Athletic, and Administrative. Individual designees, such as the football program or the biology department, can then be assigned to the appropriate Type.

To get the designation types and designees to use for a form, call the *getDesignationTypes* and *getDesignees* methods. To specify designations for a donation, use the designated.X.id and designated.X.amount parameters, substituting numbers for the "X" part of the parameter names to identify ID and amount pairs (the numbers do not need to be sequential). To specify "write-in" designations, use the designated_write_in.X.name, designated_write_in.X.contact, and designated_write_in.X.amount parameters.

Sustaining (Recurring) Gifts

Sustainer giving lets donors set up recurring, automatic gifts, with either pre-set amounts and durations or flexible donor-specified commitments. Use the sustaining.frequency and sustaining.duration parameters to make sustaining donations. The shadow donation form must contain one of the sustaining giving donation level elements.

Installment programs enable higher-dollar giving levels with payment programs that offer the benefits of large gifts but automatically pay out over the schedules you design and donors select. Use the installment.frequency and installment.duration parameters to make installment donations. The shadow donation form must contain the installment plan donation level element.

Donation Errors

If the call to the API succeeds but one or more errors are encountered during transaction processing, a donationResponse object will be returned listing reasons for the error. To correlate with the standard donation form processing, `pageError` elements in the response are the same messages that would be displayed near the top of the page and are sometimes generic. The `fieldError` elements in the response are the same messages that would be displayed near input fields throughout the page and are always specific to the validation of input for an input field.

Soft-crediting Gifts to TeamRaiser Participants, Teams, or Events

You can soft-credit donations processed through the `donate` or `startDonation` methods to TeamRaiser particpants, teams or events. In each case, you must include the TeamRaiser Event ID in the fr_id parameter. To soft-credit the gift to a participant: specify soft_credit_type as TR_PARTICIPANT, and pass the Constituent ID of the particpant in soft_credit_id. To soft-credit the gift to a team: specify soft_credit_type as TR_TEAM, and pass the Team ID in soft_credit_id. To soft-credit the gift to an event: specify soft_credit_type as TR_EVENT, and pass the Event ID (the same value as in fr_id) in soft_credit_id. The system returns an error if you attempt to soft-credit a donation to an event that is not currently accepting donations or to a team that does not accept team gifts.

Error Responses

For common API errors such as authentication errors, the common error response is returned. An XML example:

```
<errorResponse xmlns="http://convio.com/crm/v1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://convio.com/crm/v1.0 http://service.convio.net/
xmlschema/crm.public.v1.xsd">
  <code>2</code>
  <message>Incorrect API key.  Verify that the value of the parameter api_key
 matches the value of the SDP CONVIO_API_KEY.</message>
</errorResponse>
```

A JSON example:

```
{"errorResponse":{
  "code": "2",
  "message": "Incorrect API key.  Verify that the value of the parameter
 api_key matches the value of the SDP CONVIO_API_KEY.",
}}
```

For donation-specific errors, the "donationResponse" is returned and contains a list of the error messages. An XML example:

```
<donationResponse xmlns="http://convio.com/crm/v1"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://convio.com/crm/v1.0 http://service.convio.net/
xmlschema/crm.public.v1.xsd">
  <errors>
    <code>101</code>
    <message>Error: There was a problem encountered while processing your
 donation.</message>
    <reason>FIELD_VALIDATION</reason>
    <pageError>There was a problem processing your request.  Please see
 below.</pageError>
    <fieldError>An email address is required.</fieldError>
    <fieldError>Billing state or province is required.</fieldError>
    <fieldError>Billing last name is required.</fieldError>
    <fieldError>Billing zip or postal code is required.</fieldError>
    <fieldError>Billing street address is required.</fieldError>
    <fieldError>Billing first name is required.</fieldError>
    <fieldError>Billing city is required.</fieldError>
  </errors>
</donationResponse>
```

A JSON example:

```
{"donationResponse":
  {"errors":
    {"code":"101",
     "reason":"FIELD_VALIDATION",
     "message":"Error: There was a problem encountered while processing your
 donation.",
     "fieldError":["An email address is required.",
                   "Billing state or province is required.",
                   "Billing last name is required.",
                   "Billing zip or postal code is required.",
                   "Billing street address is required.",
                   "Billing first name is required.",
                   "Billing city is required."],
     "pageError":"There was a problem processing your request.  Please see
 below."}
  }
}
```

Note: If you specify "true" in the optional suppress_response_codes parameter, both success and error results will return a donationResponse document with an HTTP return code of 200 - OK. You must parse the donationResponse document for an errors element to determine whether or not the call succeeded.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

billing.address.city

Required. The billing address of the credit card owner. If not supplied, defaults to donor.address.city.

Type xsd:string.

billing.address.state

Required. The billing address of the credit card owner. If not supplied, defaults to donor.address.state.

Type xsd:string.

billing.address.street1

Required. The billing address of the credit card owner. If not supplied, defaults to donor.address.street1.

Type xsd:string.

billing.address.zip

Required. The billing address of the credit card owner. If not supplied, defaults to donor.address.zip.

Type xsd:string.

billing.name.first

Required. The billing name of the credit card owner. If not supplied, defaults to donor.name.first.

Type xsd:string.

billing.name.last

Required. The billing name of the credit card owner. If not supplied, defaults to donor.name.last.

Type xsd:string.

donor.email

Required. The donor's email address.

Type xsd:string.

form_id

Required. The ID of the donation form.

Type xsd:nonNegativeInteger.

level_id

Required. The ID of the donation level.

Type xsd:nonNegativeInteger.

ach_account

Optional. ACH (debit) account number.

Type xsd:string.

ach_account_type

Optional. ACH account type (CHECKING/SAVINGS).

Type xsd:string.

Default is CHECKING.

Options are:
- CHECKING
- SAVINGS

ach_bank

Optional. ACH bank number (Canada).

Type xsd:string.

ach_routing

Optional. ACH routing number.

Type xsd:string.

ach_transit

Optional. ACH transit number (Canada).

Type xsd:string.

additional_amount

Optional. Donation amount in addition to level amount.

Type xsd:string.

anonymous

Optional. Whether to mark the donation as "anonymous".

Type xsd:boolean.

billing.address.country

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.address.county

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.address.street2

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.address.street3

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.name.middle

Optional. The billing name of the credit card owner.

Type xsd:string.

billing.name.profSuffix

Optional. The billing name of the credit card owner.

Type xsd:string.

billing.name.suffix

Optional. The billing name of the credit card owner.

Type xsd:string.

billing.name.title

Optional. The billing name of the credit card owner.

Type xsd:string.

card_cvv

Optional. Credit card Verification Value. This is usually required, depending on the configuration of the Convio site.

Type xsd:string.

card_exp_date

Optional. Credit card expiration date in MM/YYYY format. Use either this or both card_exp_date_month and card_exp_date_year.

Type xsd:string.

card_exp_date_month

Optional. Credit card expiration month (1-12).

Type xsd:nonNegativeInteger.

card_exp_date_year

Optional. Credit card expiration year.

Type xsd:nonNegativeInteger.

card_number

Optional. Credit card number. For security, this must be specified in the form -- it is an error to include it in the URL.

Type xsd:string.

designated.X.amount

Optional. The amount of a donation to designate to a designee. The "X" part of the parameter must be an integer used to correspond to designated.X.id or, in the case of write-in designations, designated.X.name and designated.X.contact. This allows multiple pairs of ID/amount for multiple designations.

Type xsd:nonNegativeInteger.

designated.X.id

Optional. The ID of a designee to which to designate all or part of a donation. The "X" part of the parameter must be an integer used to correspond to designated.X.amount. This allows multiple pairs of ID/amount for multiple designations.

Type xsd:nonNegativeInteger.

designated_write_in.X.amount

Optional. The amount of a donation to designate to a designee. The "X" part of the parameter must be an integer used to correspond to designated_write_in.X.name and designated_write_in.X.contact. This allows multiple pairs of ID/amount for multiple designations.

Type xsd:nonNegativeInteger.

designated_write_in.X.contact

Optional. The contact information of an organization to which to designate all or part of a donation. The "X" part of the parameter must be an integer used to correspond to designated_write_in.X.name and designated_write_in.X.amount. This allows multiple sets of name/contact/amount for multiple designations.

Type xsd:string.

designated_write_in.X.name

Optional. The name of an organization to which to designate all or part of a donation. The "X" part of the parameter must be an integer used to correspond to designated_write_in.X.contact and designated_write_in.X.amount. This allows multiple sets of name/contact/amount for multiple designations.

Type xsd:string.

donor.address.city

Optional. The donor's address.

Type xsd:string.

donor.address.country

Optional. The donor's address.

Type xsd:string.

donor.address.county

Optional. The donor's address.

Type xsd:string.

donor.address.state

Optional. The donor's address.

Type xsd:string.

donor.address.street1

Optional. The donor's address.

Type xsd:string.

donor.address.street2

Optional. The donor's address.

Type xsd:string.

donor.address.street3

Optional. The donor's address.

Type xsd:string.

donor.address.zip

Optional. The donor's address.

Type xsd:string.

donor.email_opt_in

Optional. The donor's email opt-in selection.

Type xsd:string.

donor.employer

Optional. The donor's employer.

Type xsd:string.

donor.name.first

Optional. The donor's name.

Type xsd:string.

donor.name.last

Optional. The donor's name.

Type xsd:string.

donor.name.middle

Optional. The donor's name.

Type xsd:string.

donor.name.profSuffix

Optional. The donor's name.

Type xsd:string.

donor.name.suffix

Optional. The donor's name.

Type xsd:string.

donor.name.title

Optional. The donor's name.

Type xsd:string.

donor.occupation

Optional. The donor's occupation.

Type xsd:string.

donor.phone

Optional. The donor's phone number.

Type xsd:string.

donor.phone_type

Optional. The type of the donor's phone number .

Type xsd:string.

Options are:

- home
- work
- other

ecard.copy_sender

Optional. Whether to send a copy of the e-card to the sender.

Type xsd:boolean.

ecard.id

Optional. The unique ID of the pre-defined Convio e-card template to use.

Type xsd:nonNegativeInteger.

ecard.message

Optional. The message body of the e-card.

Type xsd:string.

ecard.recipients

Optional. The email addresses of intended e-card recipients (RFC-5322 format) e.g. 'John Doe<user1@host.org>, Jane Roe<user2@host.com>'.

Type xsd:string.

ecard.send

Optional. Whether or not to send an e-card recognizing the gift.

Type xsd:boolean.

Default is false.

ecard.send_date

Optional. Date to send the e-card (ISO 8601 format) e.g. '2009-12-31'.

Type xsd:date.

ecard.subject

Optional. The subject of the e-card.

Type xsd:string.

fr_id

Optional. Specifies the ID of the TeamRaiser event associated with this donation.

Type xsd:nonNegativeInteger.

gift_aid

Optional. Gift aid status (UK Only).

Type xsd:string.

installment.duration

Optional. The duration of an installment plan donation, which is the number of payments that the donation is divided into. To make an installment plan donation, both installment.frequency and installment.duration must be specified.

Type xsd:nonNegativeInteger.

installment.frequency

Optional. The frequency of an installment plan donation, which is the interval between payments. To make an installment plan donation, both installment.frequency and installment.duration must be specified.

Type crm:donationSustainingFrequency.

Options are:

- one-time
- monthly
- quarterly
- semi-annually
- annually

joint_donor.name.first

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.last

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.middle

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.suffix

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.title

Optional. The name for a joint donor.

Type xsd:string.

level_autorepeat

Optional. Whether or not the "recurring gift" option on the Donation Level shadow form should be selected. Valid only for Level shadow forms on which a recurring gift option has been defined.

Type xsd:boolean.

Default is false.

matching_eligible

Optional. Whether or not the donation is eligible for a matching gift.

Type xsd:boolean.

other_amount

Optional. User specified donation amount.

Type xsd:string.

premium_id

Optional. Unique ID of premium.

Type xsd:nonNegativeInteger.

premium_option

Optional. Premium option.

Type xsd:string.

remember_me

Optional. Whether to set a log-in cookie.

Type xsd:boolean.

send_receipt

Optional. Whether to send a receipt email to the donor. The default behavior is to send a receipt.

Type xsd:boolean.

Default is true.

send_registration_email

Optional. If a new constituent record is created for the donor, this specifies whether to send a registration "welcome" email to the donor. The default behavior is to send it.

Type xsd:boolean.

Default is true.

shipping.address.city

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.country

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.county

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.state

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.street1

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.street2

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.street3

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.zip

Optional. Shipping address for premium.

Type xsd:string.

shipping.email

Optional. Email address of the recipient.

Type xsd:string.

shipping.name.first

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.last

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.middle

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.profSuffix

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.suffix

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.title

Optional. Premium recipient's name.

Type xsd:string.

shipping.phone

Optional. Phone number of the recipient.

Type xsd:string.

soft_credit_id

Optional. Specifies the ID of the entity to which this donation will be soft credited. This will be either the event ID, team ID, or participant (constituent) ID, depending on the value specified for soft_credit_type.

Type xsd:nonNegativeInteger.

soft_credit_type

Optional. Specifies type of entity to which this donation will be soft credited. Donations that specify a soft credit type must also specify a soft_credit_id value.

Type xsd:string.

Options are:

- TR_PARTICIPANT
- TR_TEAM
- TR_EVENT
- TRIBUTE_GIFT

summary

Optional. The type of donation summary to return. There are three options:'data' returns a structured list of information about every form field; 'page' returns the HTML content (within the "body" element) of the "Thank you" page shown after donations via web page; 'both' returns both types of summary.

Type xsd:string.

Options are:

- data
- page
- both

sustaining.duration

Optional. The duration of a sustaining donation, which is the interval between payments. Specify zero to indicate that the donation should repeat indefinitely. To make a sustaining donation, both sustaining.frequency and sustaining.duration must be specified.

Type xsd:nonNegativeInteger.

sustaining.frequency

Optional. The frequency of a sustaining donation, which is how often the donation is repeated. To make a sustaining donation, both sustaining.frequency and sustaining.duration must be specified.

Type crm:donationSustainingFrequency.

Options are:

- one-time
- monthly
- quarterly
- semi-annually
- annually

teamraiser.message_to_participant

Optional. If this donation is a TeamRaiser Gift, the body of the message to send to the TeamRaiser participant.

Type xsd:string.

teamraiser.recognition_name

Optional. If this donation is a TeamRaiser Gift, the recognition name.

Type xsd:string.

teamraiser.show_gift_to_public

Optional. If this donation is a TeamRaiser Gift, whether or not to the TeamRaiser gift is publicly visible.

Type xsd:boolean.

tribute.honoree.deceased

Optional. Whether the tribute honoree is deceased. If set, it overrides "tribute.type" by setting it to "memorial" if true (deceased) and "tribute" if false (not deceased).

Type xsd:boolean.

tribute.honoree.name.first

Optional. The first name of the person to honor with the tribute. Takes precedence over tribute.honoree.name.full.

Type xsd:string.

tribute.honoree.name.full

Optional. The full name of the person to honor with the tribute. Only used if the first and last names are not specified.

Type xsd:string.

tribute.honoree.name.last

Optional. The last name of the person to honor with the tribute. Takes precedence over tribute.honoree.name.full.

Type xsd:string.

tribute.honoree.name.title

Optional. The name title of the person to honor with the tribute.

Type xsd:string.

tribute.message.body

Optional. The body of the tribute notification message.

Type xsd:string.

tribute.message.closing

Optional. The closing of the tribute notification message.

Type xsd:string.

tribute.message.include_amount

Optional. Whether to include the donation amount in the tribute notification message.

Type xsd:boolean.

tribute.message.signature

Optional. The signature of the tribute notification message.

Type xsd:string.

tribute.notify.address.city

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.country

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.county

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.state

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.street1

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.street2

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.street3

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.zip

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.name.full

Optional. The full name of the person to notify of the tribute.

Type xsd:string.

tribute.notify.name.title

Optional. The name title of the person to notify of the tribute.

Type xsd:string.

tribute.type

Optional. The type of tribute.

Type xsd:string.

Options are:

- memorial
- tribute

validate

Optional. Whether to enforce validation of every field according to the specification in the ghost form ("true" or "false"). Some validation is always in effect, such as ensuring that the campaign, form, and level IDs reference published entities and that there is sufficient information to process the transaction.

Type xsd:boolean.

Default is false.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<donationResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <donation>
    <cons_id>1006789</cons_id>
    <confirmation_code>0-0000-0-0000-0000</confirmation_code>
    <date_time>2013-02-03T05:18:27.873-06:00</date_time>
    <org_tax_id>00-0000000</org_tax_id>
    <reward_points_earned>100</reward_points_earned>
    <transaction_id>100101</transaction_id>
    <amount>
      <decimal>100</decimal>
      <formatted>$100.00</formatted>
    </amount>
    <tax_deductible_amount>
      <decimal>100</decimal>
      <formatted>$100.00</formatted>
    </tax_deductible_amount>
    <value_of_goods>
      <decimal>0</decimal>
      <formatted>$0.00</formatted>
    </value_of_goods>
    <summary_data>
      <field header="false">
        <label>Transaction Date:</label>
        <value>0/0/201</value>
        <name>date_id</name>
      </field>
      <field header="true">
        <label>Gift Information</label>
        <value/>
        <name>section-header</name>
      </field>
```

```
      <field header="false">
        <label>Amount:</label>
        <value>$100.00</value>
        <name>level_id</name>
      </field>
      <field header="true">
        <label>Billing Information</label>
        <value/>
        <name>section-header</name>
      </field>
      <field header="false">
        <label>Billing First Name:</label>
        <value>John</value>
        <name>billing.name.first</name>
      </field>
      <field header="false">
        <label>Billing Last Name:</label>
        <value>Doe</value>
        <name>billing.name.last</name>
      </field>
      <field header="true">
        <label>Payment Information</label>
        <value/>
        <name>section-header</name>
      </field>
      <field header="false">
        <label>Payment Type:</label>
        <value>Credit Card</value>
        <name>payment_type</name>
      </field>
      <field header="false">
        <label>Credit Card Number:</label>
        <value>***************1111</value>
        <name>card_number</name>
      </field>
    </summary_data>
  </donation>
</donationResponse>
```

JSON response

```
{"donationResponse":{"donation":{"amount":
{"formatted":"$100.00","decimal":"100"},"value_of_goods":
{"formatted":"$0.00","decimal":"0"},"summary_data":{"field":
[{"name":"date_id","value":"0/0/201","label":"Transaction
 Date:","header":"false"},{"name":"section-header","value":
{},"label":"Gift Information","header":"true"},
{"name":"level_id","value":"$100.00","label":"Amount:","header":"false"},
{"name":"section-header","value":{},"label":"Billing
 Information","header":"true"},
{"name":"billing.name.first","value":"John","label":"Billing First
 Name:","header":"false"},
{"name":"billing.name.last","value":"Doe","label":"Billing Last
 Name:","header":"false"},{"name":"section-header","value":{},"label":"Payment
 Information","header":"true"},{"name":"payment_type","value":"Credit
 Card","label":"Payment Type:","header":"false"},
{"name":"card_number","value":"***************1111","label":"Credit
 Card Number:","header":"false"}]},"tax_deductible_amount":
{"formatted":"$100.00","decimal":"100"},"org_tax_id":"00-0000000","confirmation_code":"0
```

getDesignationTypes

This method returns designated giving types for a Donations 2 (not "Donations Classic") donation form.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRDonationAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRDonationAPI?method=getDesignationTypes`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&form_id=`*value*]

Usage Notes

The Designated Giving feature in Online Giving allows you to set up Designees within your organization and then assign all funds from a donation form to a specific designee or permit your donors to specify which Designee(s) they would like to target with their gifts. To use this feature, you must first set up your list of Designees and Designation Types. Designation Types help organize your designees for display to your donors. For example, an educational institution might create the following Designation Types: Academic, Athletic, and Administrative. Individual designees, such as the football program or the biology department, can then be assigned to the appropriate Type.

Designation Types can be given a sort order. In the donation forms generated by the site, they are presented visually in ascending order of this value. In a custom form powered by the donation APIs, the form author may choose to either use or ignore this value when presenting them to the donors.

Designation Types that have been archived are not included in the results.

Donation forms that accept designations may be configured to either accept all designees or to only accept a specific subset of the designees. If the form accepts only a subset of designees and the validate parameter is set to "true" when calling the *donate* method, then an error will occur if the donor selects a designee that is not allowed. Use the form_id parameter for this method to restrict the results to only the designation types allowed for a specific form (those associated with the designees that are allowed for the form) to avoid this situation.

See also *getDesignees* and *donate*.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

form_id

Optional. The ID of the donation form.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getDesignationTypesResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <type>
    <id>1021</id>
    <name>Program 1</name>
    <sortOrder>2</sortOrder>
  </type>
  <type>
    <id>1022</id>
    <name>Program 2</name>
```

```
    <sortOrder>1</sortOrder>
  </type>
</getDesignationTypesResponse>
```

JSON response

```
{"getDesignationTypesResponse":{"type":
[{"id":"1021","sortOrder":"2","name":"Program 1"},
{"id":"1022","sortOrder":"1","name":"Program 2"}]}}
```

getDesignees

This method returns active designated giving designees for a Donations 2 (not "Donations Classic") donation form.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRDonationAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRDonationAPI?method=getDesignees`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&form_id=`*value*] [`&html_format=`*text* | *xhtml*]

Usage Notes

The Designated Giving feature in Online Giving allows you to set up Designees within your organization and then assign all funds from a donation form to a specific designee or permit your donors to specify which Designee(s) they would like to target with their gifts. To use this feature, you must first set up your list of Designees and Designation Types. Designation Types help organize your designees for display to your donors. For example, an educational institution might create the following Designation Types: Academic, Athletic, and Administrative. Individual designees, such as the football program or the biology department, can then be assigned to the appropriate Type.

Designees are the actual departments or programs within your organization that will receive the funds you raise. Tying a Donation Form to one or more designees is not required -- or even desirable in every organization. Often funds must be allocated strictly through a budgeting process. For some organizations, however, every gift must be designated for a specific purpose. In addition, for many organizations, presenting your donors with a list of one or more specific programs that they can "designate" as recipients of their gifts can increase the number and amount of gifts received.

Designees that have been archived are not included in the results.

Donation forms that accept designations may be configured to either accept all designees or to only accept a specific subset of the designees. If the form accepts only a subset of designees and the validate parameter is set to "true" when calling the *donate* method, then an error will occur if the donor selects a designee that is not allowed. Use the form_id parameter for this method to restrict the results to only the designees allowed for a specific form to avoid this situation.

See also *getDesignationTypes* and *donate*.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

form_id

Optional. The ID of the donation form.

Type xsd:nonNegativeInteger.

html_format

Optional. For returned data that contains HTML, this specifies whether to return it as HTML-escaped text (value = "text") or as XHTML within the XML response document (value = "xhtml"). This does not apply when a JSON response is requested.

Type xsd:token.

Default is text.

Options are:

- text
- xhtml

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getDesigneesResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <designee>
    <id>1000</id>
    <name>Program 1</name>
    <typeId>1022</typeId>
    <description>
      <text>&lt;p>The &lt;em>first&lt;/em> designated giving program.&lt;/p></
text>
    </description>
  </designee>
  <designee>
    <id>1001</id>
    <name>Program 2</name>
    <typeId>1022</typeId>
    <description>
      <text>&lt;p>The &lt;em>second&lt;/em> designated giving program.&lt;/
p></text>
    </description>
  </designee>
</getDesigneesResponse>
```

JSON response

```
{"getDesigneesResponse":{"designee":[{"id":"1000","description":
{"text":"<p>The <em>first<\/em> designated giving program.<\/
p>"},"name":"Program 1","typeId":"1022"},{"id":"1001","description":
{"text":"<p>The <em>second<\/em> designated giving program.<\/
p>"},"name":"Program 2","typeId":"1022"}]}}
```

Examples

getDonationFormInfo

This method returns the information required to submit a donation form.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRDonationAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRDonationAPI?method=getDonationFormInfo &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&form_id=`*value* [`&fr_id=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

form_id

Required. The ID of the donation form.

Type xsd:nonNegativeInteger.

fr_id

Optional. The ID of the teamraiser.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getDonationFormInfoResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <donationLevels>
    <donationLevel>
      <level_id>10</level_id>
      <name>Sample Donation Level $10</name>
      <amount>
        <decimal>0.10</decimal>
        <formatted>$10.00</formatted>
      </amount>
      <userSpecified>false</userSpecified>
    </donationLevel>
    <donationLevel>
      <level_id>20</level_id>
      <name>Sample Donation Level $20</name>
      <amount>
        <decimal>0.20</decimal>
        <formatted>$20.00</formatted>
      </amount>
      <userSpecified>false</userSpecified>
    </donationLevel>
    <donationLevel>
      <level_id>30</level_id>
      <name>Sample Donation Level $30</name>
      <amount>
        <decimal>0.30</decimal>
        <formatted>$30.00</formatted>
      </amount>
      <userSpecified>false</userSpecified>
    </donationLevel>
  </donationLevels>
  <paymentCards>
    <paymentCard>Visa</paymentCard>
    <paymentCard>MasterCard</paymentCard>
    <paymentCard>American Express</paymentCard>
```

```
    <paymentCard>Discover</paymentCard>
  </paymentCards>
  <bankTransfers>true</bankTransfers>
  <externalProcessors>
    <externalProcessor>PayPal</externalProcessor>
    <externalProcessor>Amazon</externalProcessor>
  </externalProcessors>
  <donationFields>
    <donationField>
      <elementName>form_id</elementName>
      <required>true</required>
      <defaultValue>0</defaultValue>
    </donationField>
    <donationField>
      <elementName>level_id</elementName>
      <required>true</required>
      <defaultValue/>
    </donationField>
    <donationField>
      <elementName>billing.name.first</elementName>
      <required>true</required>
      <defaultValue/>
    </donationField>
    <donationField>
      <elementName>billing.name.last</elementName>
      <required>true</required>
      <defaultValue/>
    </donationField>
    <donationField>
      <elementName>billing.address.street1</elementName>
      <required>true</required>
      <defaultValue/>
    </donationField>
    <donationField>
      <elementName>billing.address.city</elementName>
      <required>true</required>
      <defaultValue/>
    </donationField>
    <donationField>
      <elementName>billing.address.state</elementName>
      <required>true</required>
      <defaultValue/>
    </donationField>
    <donationField>
      <elementName>billing.address.zip</elementName>
      <required>true</required>
      <defaultValue/>
    </donationField>
    <donationField>
      <elementName>donor.email</elementName>
      <required>true</required>
      <defaultValue/>
    </donationField>
  </donationFields>
  <supportsDesignation>false</supportsDesignation>
  <supportsWriteInDesignation>false</supportsWriteInDesignation>
  <supportsEcard>false</supportsEcard>
  <supportsInstallments>false</supportsInstallments>
  <supportsSustaining>false</supportsSustaining>
  <supportsLevelAutorepeat>false</supportsLevelAutorepeat>
  <supportsGiftAid>false</supportsGiftAid>
  <supportsTribute>false</supportsTribute>
  <teamraiserTeamGiftsAllowed>false</teamraiserTeamGiftsAllowed>
</getDonationFormInfoResponse>
```

JSON response


```
{"getDonationFormInfoResponse":
{"supportsLevelAutorepeat":"false","donationLevels":{"donationLevel":
[{"amount":{"formatted":"$10.00","decimal":"0.10"},"name":"Sample
 Donation Level $10","level_id":"10","userSpecified":"false"},{"amount":
{"formatted":"$20.00","decimal":"0.20"},"name":"Sample Donation
 Level $20","level_id":"20","userSpecified":"false"},{"amount":
{"formatted":"$30.00","decimal":"0.30"},"name":"Sample Donation Level
 $30","level_id":"30","userSpecified":"false"}]},"teamraiserTeamGiftsAllowed":"false","pa
{"paymentCard":["Visa","MasterCard","American
 Express","Discover"]},"externalProcessors":{"externalProcessor":
["PayPal","Amazon"]},"supportsGiftAid":"false","supportsTribute":"false","supportsEcard"
{"donationField":
[{"elementName":"form_id","defaultValue":"0","required":"true"},
{"elementName":"level_id","defaultValue":{},"required":"true"},
{"elementName":"billing.name.first","defaultValue":{},"required":"true"},
{"elementName":"billing.name.last","defaultValue":{},"required":"true"},
{"elementName":"billing.address.street1","defaultValue":
{},"required":"true"},{"elementName":"billing.address.city","defaultValue":
{},"required":"true"},{"elementName":"billing.address.state","defaultValue":
{},"required":"true"},{"elementName":"billing.address.zip","defaultValue":
{},"required":"true"},{"elementName":"donor.email","defaultValue":
{},"required":"true"}]},"supportsWriteInDesignation":"false","supportsInstallments":"fals
```


getRecurringConfiguration

Returns the configuration available for a specific recurring gift.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRRecurringAPI	SRRecurringAPI	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRRecurringAPI?method=getRecurringConfiguration &recurring_gift_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRRecurringAPI?method=getRecurringConfiguration & recurring_gift_id =`*value* `& cons_id =`*value*

Usage Notes

This method returns a list of the actions allowed for a recuring gift and other information about the configuration of the Convio Gift Service Center.

You may use the following methods to allow a logged-in user to manage their recurring gifts themselves via the Client API or to manage recurring gifts as an adminstrator via the Server API:

- *calculateNextPaymentDate*
- *cancelRecurringGift*
- *getRecurringConfiguration*
- *getRecurringGiftDetails*
- *getRecurringGifts*
- *getRecurringHistory*
- *skipNextPayment*

- *updateRecurringBankAccountInfo*
- *updateRecurringCreditCardInfo*
- *updateRecurringGift_method*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

cons_id

Required. Specifies the ID of the constiuent.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getRecurringConfigurationResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <allowModifyBillingInformation>true</allowModifyBillingInformation>
  <allowDiscontinuePayments>true</allowDiscontinuePayments>
  <allowSkipNextPayment>true</allowSkipNextPayment>
  <allowModifyGiftAmount>true</allowModifyGiftAmount>
  <allowModifyNextPaymentDate>true</allowModifyNextPaymentDate>
  <showTopNavBar>true</showTopNavBar>
  <htmlContent>&lt;p>content&lt;/p></htmlContent>
  <allowCCOption>true</allowCCOption>
  <allowACHOption>true</allowACHOption>
  <allowACHCanadaOption>false</allowACHCanadaOption>
  <enableCardCvvVerification>true</enableCardCvvVerification>
  <cvvHelpLink>http://www.example.com</cvvHelpLink>
</getRecurringConfigurationResponse>
```

JSON response

```
{"getRecurringConfigurationResponse":
{"enableCardCvvVerification":"true","allowSkipNextPayment":"true","showTopNavBar":"true",
\/
p>","allowModifyGiftAmount":"true","allowCCOption":"true","allowACHCanadaOption":"false",
www.example.com","allowDiscontinuePayments":"true","allowModifyNextPaymentDate":"true","a
```

getRecurringGiftDetails

Returns the details of the specified recurring gift.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRRecurringAPI	SRRecurringAPI	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRRecurringAPI?method=getRecurringGiftDetails &recurring_gift_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRRecurringAPI?method=getRecurringGiftDetails & recurring_gift_id =`*value* `& cons_id =`*value*

Usage Notes

This method returns information about a recurring gift, including last payement and next payment dates, amount, duration and frequency, payment card and billing information.

You may use the following methods to allow a logged-in user to manage their recurring gifts themselves via the Client API or to manage recurring gifts as an adminstrator via the Server API:

- *calculateNextPaymentDate*
- *cancelRecurringGift*
- *getRecurringConfiguration*
- *getRecurringGiftDetails*
- *getRecurringGifts*
- *getRecurringHistory*
- *skipNextPayment*
- *updateRecurringBankAccountInfo*
- *updateRecurringCreditCardInfo*
- *updateRecurringGift_method*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

cons_id

Required. Specifies the ID of the constiuent.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getRecurringGiftDetailsResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <recurringGiftDetails>
    <id>1</id>
    <tenderType>CREDIT_CARD</tenderType>
    <creditInfo>
      <maskedCreditCardNumber>************1234</maskedCreditCardNumber>
      <expirationMonth>2</expirationMonth>
      <expirationYear>2013</expirationYear>
    </creditInfo>
    <nextPaymentDate>2013-02-03-06:00</nextPaymentDate>
    <previousPaymentDate>2013-02-03-06:00</previousPaymentDate>
    <amount>
      <decimal>10.00</decimal>
      <formatted>$10.00</formatted>
    </amount>
    <frequency>monthly</frequency>
    <billingName>
      <first>John</first>
      <last>Doe</last>
    </billingName>
    <billingAddress>
      <street1>11500 Fake Street</street1>
      <city>New York City</city>
      <state>NY</state>
      <zip>12345</zip>
    </billingAddress>
  </recurringGiftDetails>
</getRecurringGiftDetailsResponse>
```

JSON response

```
{"getRecurringGiftDetailsResponse":{"recurringGiftDetails":{"amount":
{"formatted":"$10.00","decimal":"10.00"},"id":"1","previousPaymentDate":"2013-02-03-06:00
{"zip":"12345","street1":"11500 Fake Street","state":"NY","city":"New
 York City"},"frequency":"monthly","creditInfo":
{"expirationYear":"2013","expirationMonth":"2","maskedCreditCardNumber":"************1234
{"last":"Doe","first":"John"}}}}
```

getRecurringGifts

Returns the recurring gifts associated with a single constituent.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRRecurringAPI	SRRecurringAPI	GET, POST	Yes.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRRecurringAPI?method=getRecurringGifts`[`&auth=`*value*][`&sso_auth_token=`*value*]
Server API Syntax
`https://`*secure2.convio.net/organization*`/site/SRRecurringAPI?method=getRecurringGifts &cons_id =`*value*

Usage Notes

This method returns the list of recurring gifts for a specified constituent.

You may use the following methods to allow a logged-in user to manage their recurring gifts themselves via the Client API or to manage recurring gifts as an adminstrator via the Server API:

- *calculateNextPaymentDate*
- *cancelRecurringGift*
- *getRecurringConfiguration*
- *getRecurringGiftDetails*
- *getRecurringGifts*
- *getRecurringHistory*
- *skipNextPayment*
- *updateRecurringBankAccountInfo*
- *updateRecurringCreditCardInfo*
- *updateRecurringGift_method*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Required. Specifies the ID of the constiuent.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getRecurringGiftsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <recurringGift>
    <tenderType>CREDIT_CARD</tenderType>
    <title>fake title</title>
    <id>1</id>
    <creditInfo>
      <maskedCreditCardNumber>************1234</maskedCreditCardNumber>
      <expirationMonth>2</expirationMonth>
      <expirationYear>2013</expirationYear>
    </creditInfo>
    <nextPaymentDate>2013-02-03-06:00</nextPaymentDate>
    <previousPaymentDate>2013-02-03-06:00</previousPaymentDate>
    <amount>
      <decimal>10.00</decimal>
      <formatted>$10.00</formatted>
    </amount>
    <frequency>monthly</frequency>
    <tenderInstance>MASTER</tenderInstance>
  </recurringGift>
</getRecurringGiftsResponse>
```

JSON response


```
{"getRecurringGiftsResponse":{"recurringGift":
{"tenderInstance":"MASTER","amount":
{"formatted":"$10.00","decimal":"10.00"},"id":"1","title":"fake
 title","previousPaymentDate":"2013-02-03-06:00","nextPaymentDate":"2013-02-03-06:00","fr
{"expirationYear":"2013","expirationMonth":"2","maskedCreditCardNumber":"************1234
```


getRecurringHistory

Retrieves payment history on a recurring gift for the given calendar year.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRRecurringAPI	SRRecurringAPI	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRRecurringAPI?method=getRecurringHistory &recurring_gift_id=`*value* `&sustainer_history_year=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRRecurringAPI?method=getRecurringHistory & recurring_gift_id =`*value* `& sustainer_history_year =`*value* `& cons_id =`*value*

Usage Notes

This method returns the history of payments credited against a recurring gift.

You may use the following methods to allow a logged-in user to manage their recurring gifts themselves via the Client API or to manage recurring gifts as an adminstrator via the Server API:

- *calculateNextPaymentDate*
- *cancelRecurringGift*
- *getRecurringConfiguration*
- *getRecurringGiftDetails*
- *getRecurringGifts*
- *getRecurringHistory*
- *skipNextPayment*
- *updateRecurringBankAccountInfo*
- *updateRecurringCreditCardInfo*
- *updateRecurringGift_method*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

sustainer_history_year

Required. The year for which the recurring gift history should be retrieved.

Type xsd:int.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

sustainer_history_year

Required. The year for which the recurring gift history should be retrieved.

Type xsd:int.

cons_id

Required. Specifies the ID of the constiuent.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getRecurringHistoryResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <recurringHistory>
    <amount>
      <decimal>1200</decimal>
      <formatted>12.00</formatted>
    </amount>
    <date>2013-02-03-06:00</date>
    <tenderType>CREDIT_CARD</tenderType>
    <maskedCreditCardNumber>************1111</maskedCreditCardNumber>
    <status>COMPLETED</status>
  </recurringHistory>
</getRecurringHistoryResponse>
```

JSON response

```
{"getRecurringHistoryResponse":{"recurringHistory":{"amount":
{"formatted":"12.00","decimal":"1200"},"status":"COMPLETED","maskedCreditCardNumber":"***
```

recordRecurringTransaction

This method records recurring transactions that was done outside of COM but only if its initial transactions was processed within COM.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRDonationAPI	POST	No.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRDonationAPI?method=recordRecurringTransaction&record_recurring_transaction=`*value*

Usage Notes

This Server API method permits an external processor to record payments against one or more Sustaining Gifts or pledges.

API Administrator accounts that access this API must be assigned the Offline Donation API permission type with permission to "Add Offline Donations via API". This permission is not assigned to the built-in "API Administrators" group by default.

Input data is passed url-encoded as the value of the record_recurring_transaction parameter in the following format (the format is defined in the *Donation API XML Schema*):

```
<recordRecurringTransactionRequest xmlns="http://convio.com/crm/v1.0">
    <recurPledgeRecord>
        <seed_transaction_id>23118</seed_transaction_id>
        <recur_amount>5000</recur_amount>
        <recur_date>2010-05-30T09:30:10-06:00</recur_date>
        <recur_next_payment_date>2010-06-30<recur_next_payment_date>
        <recur_success>true</recur_success>
        <send_autoresponder>false</send_autoresponder>
    </recurPledgeRecord>
    <recurPledgeRecord>
        <processor_ref_id>02RYXSPGD7WD4</processor_ref_id>
        <recur_amount>7500</recur_amount>
    </recurPledgeRecord>
</recordRecurringTransactionRequest>
```

Callers can specify up to 200 transactions per call. For each payment transaction, the caller must specify a unique identifier for the initial payment transaction in the series: either seed_transaction_id or processor_ref_id, and the recur_amount, the amount of the transaction in the smallest denomination (cents). Optionally, the caller can specify other parameters as described below.

seed_transaction_id

Optional. The Convio primary key of the initial payment record. Either this or processor_ref_id must be specified for each transaction.

Type: xsd:nonNegativeInteger

processor_ref_id

Optional. The payment processor's reference ID for the initial payment record. Either this or seed_transaction_id must be specified for each transaction.

Type: xsd:string

recur_amount

Required. Transaction value for this recurring payment in cents.

Type: xsd:integer

recur_date

Optional. Date on which the payment was processed. Default is the current server time.

Type: xsd:date

recur_next_payment_date

Optional. Expected date of next payment. Default is none.

Type: xsd:date

recur_success

Optional. Whether to record the transaction as a successful payment or a failed transaction. Default is true (successful).

Type: xsd:boolean

send_autoresponder

Optional. Whether to send an autoresponder email for this transaction to the constituent. Default is false.

Type: xsd:boolean

If there are structural errors in the request XML or if one of the transaction IDs cannot be found, no transactions will be prodessed and an error will be returned. If the XML request can be processed, a success or error result will be returned for each transaction. For example, a transaction posted against a pledge for which all the recurring payments have been made will return an error for the transaction, but will not interrupt processing of other transactions in the batch.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

record_recurring_transaction

Required. XML document specifying the recurring transactions to record.

Type crm:recordRecurringTransactionList.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<recordRecurringTransactionResponse xsi:schemaLocation="http://convio.com/
crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <transactionRecordResult>
    <success>true</success>
    <seed_transaction_id>1001</seed_transaction_id>
    <processor_ref_id>T1001</processor_ref_id>
    <transaction_id>1021</transaction_id>
  </transactionRecordResult>
</recordRecurringTransactionResponse>
```

JSON response

```
{"recordRecurringTransactionResponse":{"transactionRecordResult":
{"processor_ref_id":"T1001","seed_transaction_id":"1001","success":"true","transaction_id
```

refundOfflineDonation

This method processes a refund of a donation by a constituent to a Donations 2 (not "Donations Classic") donation form.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRDonationAPI	POST	No.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRDonationAPI?method=refundOfflineDonation` & `api_key` =*value* & `login_name` =*value* & `login_password` =*value* & `v` =*value* [& `center_id` =*value*] [& `response_format` =*xml* |*json*] [& `source` =*value*] [& `sub_source` =*value*] [&

`suppress_response_codes` =*value*] & `refund_user_confirm_code` =*value* [& `refund_amount` =*value*] [& `send_auto_responder` =*value*]

Usage Notes

This method is used to fully or partially refund offline donation transactions logged through the *addOfflineDonation* method.

It must reference a previously-created offline donation by that transaction's confirmation code. which is returned in the response from the `addOfflineDonation` method. The confirmation code is passed to `refundOfflineDonation` via the refund_user_confirm_code parameter.

Restrictions

Because a third-third party service provider may implement this API for multiple clients, you should strictly limit the permissions of the API Administrator account that has access to this API.

API Administrator accounts that access this API must be assigned the Offline Donation API permission type with permission to "Add and Refund Offline Donations via API".

Note: API Administrator accounts are limited to API access, they are never permitted to log in to Convio.

Any given API Administrator account is restricted to only refunding the offline transactions added through the *addOfflineDonation* method by the same account. An API user may not fully or partially refund transactions created online through the *donate* method, or transactions created by another Administrative user, API user, or constituent.

An API user may not refund more than the initial transaction value. The optional refund_amount is used to specify the dollar amount to refund. If the value specified is less than the current transaction value a partial refund for the amount specified is processed. If the refund_amount parameter value is not specified, is equal to 0, or is equal to the current transaction value a full refund is processed.

Error Responses

For common API errors such as authentication errors, the common error response is returned. An XML example:

```
<errorResponse xmlns="http://convio.com/crm/v1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://convio.com/crm/v1.0 http://service.convio.net/
xmlschema/crm.public.v1.xsd">
  <code>2</code>
  <message>Incorrect API key.  Verify that the value of the parameter api_key
 matches the value of the SDP CONVIO_API_KEY.</message>
</errorResponse>
```

A JSON example:

```
{"errorResponse":{
  "code": "2",
  "message": "Incorrect API key.  Verify that the value of the parameter
 api_key matches the value of the SDP CONVIO_API_KEY.",
}}
```

For refund-specific errors, the "offlineRefundResponse" is returned and contains a list of the error messages. An XML example:

```
<offlineRefundResponse xmlns="http://convio.com/crm/v1"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
   xsi:schemaLocation="http://convio.com/crm/v1.0 http://service.convio.net/
xmlschema/crm.public.v1.xsd">
 <errors>
  <code>108</code>
  <message>
   Error: Could not find transaction corresponding to user confirmation code
 '1-1001-1-1001-1200'
  </message>
  <reason>UNSPECIFIED</reason>
 </errors>
</offlineRefundResponse>
```

A JSON example:

```
{"offlineRefundResponse":
 {"errors":
  {"code":"108",
   "reason":"UNSPECIFIED",
    "message":"Error: Could not find transaction corresponding to user
 confirmation code '1-1001-1-1001-12000'"}
 }
}
```

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

refund_user_confirm_code

Required. Confirmation code of the transaction to be refunded

Type xsd:string.

refund_amount

Optional. Amount to be refunded. If refund_amount is less than the initial transaction value a partial refund will be applied. If refund_amount is the same as the initial transaction value, is 0, or is not specified a full refund will be applied.

Type crm:monetaryAmount.

send_auto_responder

Optional. Send auto responder informing donor about refund amount.

Type xsd:boolean.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<offlineRefundResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <refund>
    <info>refund info</info>
    <is_full_refund>false</is_full_refund>
    <initial_amount>
      <decimal>100.00</decimal>
      <formatted>$100.00</formatted>
    </initial_amount>
```

```
    <refund_amount>
      <decimal>75.00</decimal>
      <formatted>$75.00</formatted>
    </refund_amount>
    <final_amount>
      <decimal>25.00</decimal>
      <formatted>$25.00</formatted>
    </final_amount>
  </refund>
</offlineRefundResponse>
```

JSON response

```
{"offlineRefundResponse":{"refund":{"is_full_refund":"false","final_amount":
{"formatted":"$25.00","decimal":"25.00"},"initial_amount":
{"formatted":"$100.00","decimal":"100.00"},"refund_amount":
{"formatted":"$75.00","decimal":"75.00"},"info":"refund info"}}}
```

refundTransaction

This method processes and/or records refunds for transactions that were processed through COM.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRDonationAPI	POST	No.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRDonationAPI?method=refundTransaction & refund_transaction_request =`*value*

Usage Notes

This Server API method permits an external processor to record refunds for one or more transactions and optionally request that Convio process the refunds through its payment gateway.

API Administrator accounts that access this API must be assigned the Offline Donation API permission type with permission to "Add and Refund Offline Donations via API". This permission is not assigned to the built-in "API Administrators" group by default.

Input data is passed url-encoded as the value of the refund_request parameter in the following format (the format is defined in the *Donation API XML Schema*):

```
    <refundRequest xmlns="http://convio.com/crm/v1.0">
      <donRefundInfoList>
        <donRefundInfo>
          <transaction_id>5579</transaction_id>
          <processor_transaction_id>02RYXSPGD7WD4</processor_transaction_id>
          <user_confirmation_code>709-1201-1-5470-5579</
user_confirmation_code>
          <process_transaction>false</process_transaction>
          <refund_type>full</refund_type>
          <send_autoresponder>false</send_autoresponder>
          <refund_amount>10000</refund_amount>
        </donRefundInfo>
        <donRefundInfo>
          <user_confirmation_code>709-1201-1-4215-5508</
user_confirmation_code>
        </donRefundInfo>
```

```
      <donRefundInfoList>
      <trRefundInfoList>
        <trRefundInfo>
          <transaction_id>5510</transaction_id>
          <processor_transaction_id>02MCM1XXV1ID1</processor_transaction_id>
          <user_confirmation_code>709-1201-1-0258-5510</
user_confirmation_code>
          <process_transaction>false</process_transaction>
          <refund_type>partial</refund_type>
          <send_autoresponder>false</send_autoresponder>
          <regRefundDataList>
            <regRefundData>
              <registration_id>1001</registration_id>
              <reg_refund_amount>1000</reg_refund_amount>
              <gift_refund_amount>1300</gift_refund_amount>
            </regRefundData>
          </regRefundDataList>
        </trRefundInfo>
        <trRefundInfo>
          <transaction_id>5511</transaction_id>
          <refund_type>full</refund_type>
        </trRefundInfo>
      </trRefundInfoList>
      <pfRefundInfoList>
        <pfRefundInfo>
          <transaction_id>5422</transaction_id>
          <processor_transaction_id>02LCV1DGC0WX1</processor_transaction_id>
          <user_confirmation_code>709-1201-1-3532-5422</
user_confirmation_code>
          <process_transaction>false</process_transaction>
          <refund_type>partial</refund_type>
          <send_autoresponder>false</send_autoresponder>
          <gift_refund_amount>10000</gift_refund_amount>
        </pfRefundInfo>
        <pfRefundInfo>
          <user_confirmation_code>709-1201-1-9872-5429</
user_confirmation_code>
        </pfRefundInfo>
      </pfRefundInfoList>
      <ticketRefundInfoList>
        <ticketRefundInfo>
          <transaction_id>5589</transaction_id>
          <processor_transaction_id>01PNXSPGC8QA5</processor_transaction_id>
          <user_confirmation_code>709-1201-1-5470-5589</
user_confirmation_code>
          <process_transaction>false</process_transaction>
          <refund_type>partial</refund_type>
          <send_autoresponder>false</send_autoresponder>
          <gift_refund_amount>10000</gift_refund_amount>
          <cancel_all_tickets>false</cancel_all_tickets>
          <ticket_refund_item>
            <ticket_class_id>101</ticket_class_id>
            <ticket_refund_amount>6500</ticket_refund_amount>
            <num_of_tickets>1<num_of_tickets>
          </ticket_refund_item>
        <ticketRefundInfo>
        <ticketRefundInfo>
          <transaction_id>5590</transaction_id>
          <cancel_all_tickets>true</cancel_all_tickets>
        <ticketRefundInfo>
      </ticketRefundInfoList>
      <ecommerceRefundInfoList>
        <ecommerceRefundInfo>
          <transaction_id>5599</transaction_id>
```

```
          <processor_transaction_id>02TBPXLBC4PC5</processor_transaction_id>
          <user_confirmation_code>709-1201-1-4815-5599</
user_confirmation_code>
          <process_transaction>false</process_transaction>
          <refund_type>partial</refund_type>
          <send_autoresponder>false</send_autoresponder>
          <gift_refund_amount>10000</gift_refund_amount>
          <shipping_refund_amount>795</shipping_refund_amount>
          <refund_item_id>502</refund_item_id>
        </ecommerceRefundInfo>
        <ecommerceRefundInfo>
          <processor_transaction_id>02LWZAMGA9ID3</processor_transaction_id>
        </ecommerceRefundInfo>
    </refundRequest>
```

Callers can specify up to 200 transactions per API call. For each refund transaction, the caller must specify a unique identifier for the payment transaction to be refunded: one of transaction_id, processor_transaction_id, or user_confirmation_code.

Optionally, the caller can specify other parameters as described below. The following parameters are common to all transaction types:

transaction_id

Optional. The Convio primary key identifying the transaction being refunded. For each transaction, you must supply one unique ID: transaction_id, processor_transaction_id, or user_confirmation_code.

Type: xsd:nonNegativeInteger

processor_transaction_id

Optional. The unique transaction identifier returned by the credit card processor. For each transaction, you must supply one unique ID.

Type: xsd:string

user_confirmation_code

Optional. The user confirmation code uniquely identifying the transaction to be refunded. For each transaction, you must supply one unique ID.

Type: xsd:string

process_transaction

Optional. Boolean flag indicating whether Convio should attempt to process the refund transaction before recording it. Default is false.

Type: xsd:boolean

refund_type

Optional. Refund type. If you are refunding full amount, value should be "full" otherwise "partial". Default is "full". If you specify "partial," you must also specify at least one amount.

Type: xsd:string

Options: full, partial

send_autoresponder

Optional. Whether to send an autoresponder email for this transaction to the constituent. Default is false (do not send).

Type: xsd:boolean

The gift_refund_amount element may be used for Personal Fundraising `<pfRefundInfo>`, Ticketed Event `<ticketRefundInfo>`, and Ecommmerce Purchase `<ecommerceRefundInfo>` transaction types to specify a partial refund of an additional gift amount:

gift_refund_amount

Optional. The amount of the additional gift to be refunded in the smallest denomination (cents).

Type: xsd:nonNegativeInteger

The refund_amount element may be used for Donation `<donRefundInfo>`, TeamRaiser `<trRefundInfo>`, and Personal Fundraising `<pfRefundInfo>` transaction types to specify a partial refund:

refund_amount

Optional. The amount to be refunded in the smallest denomination (cents).

Type: xsd:nonNegativeInteger

For partial refunds of TeamRaiser `<trRefundInfo>` transaction types, you must specify either the refund_amount element or a `<regRefundDataList>` structure. The `<regRefundDataList>` structure can contain one or more `<regRefundData>` elements indicating registrations to be refunded. Note that refunding a registration fee does not cancel the constituent's registration for the event. Each `<regRefundData>` element can include the following information:

registration_id

Optional. Primary key of the associated team registration record.

Type: xsd:nonNegativeInteger

reg_refund_amount

Optional. The registration amount to refund (in cents).

Type: xsd:nonNegativeInteger

gift_refund_amount

Optional. The amount to be refunded of any additional gift made in conjunction with the registration (in cents).

Type: xsd:nonNegativeInteger

If TeamRaiser Upsells are configured for the TeamRaiser Event and included in the registration transaction, specify an `<addOnRefundDataList>` containing one or more `<addOnRefundData>` elements to identify each Upsell to refund with the following elements:

addon_id

Optional. Primary key of the associated Upsell record.

Type: xsd:nonNegativeInteger

addon_name

Optional. Upsell name.

Type: xsd:string

addon_refund_amount

Optional. Upsell amount to refund (in cents).

Type: xsd:nonNegativeInteger

For Ticketed Event `<ticketRefundInfo>` transaction types, the following parameters also apply:

cancel_all_tickets

Optional. Whether or not to cancel all tickets purchaesd through the refunded transaction. Default is false.

Type: xsd:boolean

You also can refund individual tickets by listing them in one or more `<ticket_refund_item>` nodes:

ticket_class_id

The Convio primary key of the ticket class.

Type: xsd:nonNegativeInteger

ticket_refund_amount

The amount to refund for tickets of this class (in cents).

Type: xsd:nonNegativeInteger

num_of_tickets

The number of tickets of this class to cancel.

Type: xsd:nonNegativeInteger

For Ecommerce Purchase `<ecommerceRefundInfo>` transaction types, the following parameters also apply:

shipping_refund_amount

Optional. The shipping amount to refund (in cents).

Type: xsd:nonNegativeInteger

refund_item_id

Optional. The Convio primary key of the ecommerce item refunded. A transaction may specify more than one refund_item_id.

Type: xsd:nonNegativeInteger

Note: You may not use this API to refund the seed (initial) transaction in an active series of recurring payments if subsequent payments have been made. To refund a seed transaction, an adminstrator or the constituent must first cancel the recurring gift, see also *cancelRecurringGift*.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

refund_transaction_request

Required. XML document specifying refund transactions to be processed and(or) recorded.

Type crm:refundRequest.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<refundResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <refundResultList>
    <refundResult>
      <full_refund>false</full_refund>
```

```
      <initial_amount>
        <decimal>2000</decimal>
        <formatted>$10.00</formatted>
      </initial_amount>
      <final_amount>
        <decimal>1000</decimal>
        <formatted>$10.00</formatted>
      </final_amount>
      <refund_amount>
        <decimal>1000</decimal>
        <formatted>$20.00</formatted>
      </refund_amount>
      <payment_processor_transaction_id>123</payment_processor_transaction_id>
      <payment_transaction_id>1000</payment_transaction_id>
      <refund_transaction_id>1001</refund_transaction_id>
      <success>true</success>
    </refundResult>
  </refundResultList>
</refundResponse>
```

JSON response

```
{"refundResponse":{"refundResultList":{"refundResult":
{"payment_transaction_id":"1000","refund_transaction_id":"1001","full_refund":"false","fi
{"formatted":"$10.00","decimal":"1000"},"initial_amount":
{"formatted":"$10.00","decimal":"2000"},"refund_amount":
{"formatted":"$20.00","decimal":"1000"},"payment_processor_transaction_id":"123","success
```

startDonation

This method initiates processing of a donation through an external payment processor (PayPal or Amazon).

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRDonationAPI	None.	POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRDonationAPI?method=startDonation`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*
] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [
`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*
] `&billing.address.city=`*value* `&billing.address.state=`*value*
`&billing.address.street1=`*value* `&billing.address.zip=`*value* `&billing.name.first=`*value*
`&billing.name.last=`*value* `&donor.email=`*value* `&extproc=`*amazon* | *paypal*
`&form_id=`*value* `&level_id=`*value* [`&ach_account_type=`*CHECKING* | *SAVINGS*
] [`&ach_bank=`*value*] [`&ach_transit=`*value*] [`&additional_amount=`*value*] [
`&anonymous=`*value*] [`&billing.address.country=`*value*] [`&billing.address.county=`*value*
] [`&billing.address.street2=`*value*] [`&billing.address.street3=`*value*
] [`&billing.name.middle=`*value*] [`&billing.name.profSuffix=`*value*] [
`&billing.name.suffix=`*value*] [`&billing.name.title=`*value*] [`&designated.X.amount=`*value*
] [`&designated.X.id=`*value*] [`&designated_write_in.X.amount=`*value*] [
`&designated_write_in.X.contact=`*value*] [`&designated_write_in.X.name=`*value*
] [`&donor.address.city=`*value*] [`&donor.address.country=`*value*]
[`&donor.address.county=`*value*] [`&donor.address.state=`*value*] [
`&donor.address.street1=`*value*] [`&donor.address.street2=`*value*] [
`&donor.address.street3=`*value*] [`&donor.address.zip=`*value*] [`&donor.email_opt_in=`*value*
] [`&donor.employer=`*value*] [`&donor.name.first=`*value*] [`&donor.name.last=`*value*] [

`&donor.name.middle=`*value*][`&donor.name.profSuffix=`*value*][`&donor.name.suffix=`*value*][`&donor.name.title=`*value*][`&donor.occupation=`*value*][`&donor.phone=`*value*][`&donor.phone_type=`*home* | *work* | *other*][`&ecard.copy_sender=`*value*][`&ecard.id=`*value*][`&ecard.message=`*value*][`&ecard.recipients=`*value*][`&ecard.send=`*value*][`&ecard.send_date=`*value*][`&ecard.subject=`*value*][`&finish_error_redirect=`*value*][`&finish_success_redirect=`*value*][`&fr_id=`*value*][`&gift_aid=`*value*][`&installment.duration=`*value*][`&installment.frequency=`*one-time* | *monthly* | *quarterly* | *semi-annually* | *annually*][`&joint_donor.name.first=`*value*][`&joint_donor.name.last=`*value*][`&joint_donor.name.middle=`*value*][`&joint_donor.name.suffix=`*value*][`&joint_donor.name.title=`*value*][`&level_autorepeat=`*value*][`&matching_eligible=`*value*][`&other_amount=`*value*][`&premium_id=`*value*][`&premium_option=`*value*][`&remember_me=`*value*][`&send_receipt=`*value*][`&send_registration_email=`*value*][`&shipping.address.city=`*value*][`&shipping.address.country=`*value*][`&shipping.address.county=`*value*][`&shipping.address.state=`*value*][`&shipping.address.street1=`*value*][`&shipping.address.street2=`*value*][`&shipping.address.street3=`*value*][`&shipping.address.zip=`*value*][`&shipping.email=`*value*][`&shipping.name.first=`*value*][`&shipping.name.last=`*value*][`&shipping.name.middle=`*value*][`&shipping.name.profSuffix=`*value*][`&shipping.name.suffix=`*value*][`&shipping.name.title=`*value*][`&shipping.phone=`*value*][`&soft_credit_id=`*value*][`&soft_credit_type=`*TR_PARTICIPANT* | *TR_TEAM* | *TR_EVENT* | *TRIBUTE_GIFT*][`&summary=`*data* | *page* | *both*][`&sustaining.duration=`*value*][`&sustaining.frequency=`*one-time* | *monthly* | *quarterly* | *semi-annually* | *annually*][`&teamraiser.message_to_participant=`*value*][`&teamraiser.recognition_name=`*value*][`&teamraiser.show_gift_to_public=`*value*][`&tribute.honoree.deceased=`*value*][`&tribute.honoree.name.first=`*value*][`&tribute.honoree.name.full=`*value*][`&tribute.honoree.name.last=`*value*][`&tribute.honoree.name.title=`*value*][`&tribute.message.body=`*value*][`&tribute.message.closing=`*value*][`&tribute.message.include_amount=`*value*][`&tribute.message.signature=`*value*][`&tribute.notify.address.city=`*value*][`&tribute.notify.address.country=`*value*][`&tribute.notify.address.county=`*value*][`&tribute.notify.address.state=`*value*][`&tribute.notify.address.street1=`*value*][`&tribute.notify.address.street2=`*value*][`&tribute.notify.address.street3=`*value*][`&tribute.notify.address.zip=`*value*][`&tribute.notify.name.full=`*value*][`&tribute.notify.name.title=`*value*][`&tribute.type=`*memorial* | *tribute*][`&validate=`*value*]

Usage Notes

This method allows you to process a donation through Amazon or PayPal. To use it, you must have a gateway to your Amazon or PayPal account set up in your Convio merchant account configuration. Also, the shadow donation form you reference by the form_id parameter must allow payments through the external payment processor gateway.

For the most part, the `startDonation` method takes the same input parameters, functions the same way, and returns the same response data as the *donate* method. It supports the same types of transactions. There are a few differences:

- The extproc parameter is required. You must specify either "amazon" or "paypal."
- The ach_account, ach_routing, card_cvv, card_exp_date, card_exp_date_month, card_exp_date_year, and card_number fields, if specified, are ignored, since payment information is maintained by the external payment processor.
- You may optionally provide finish_success_redirect and finish_error_redirect URLs to guide page flow after returning from the external processor's site.

The external payment processor may return billing and/or shipping address information for the transaction, in which case the returned data will override any information you provided when calling startDonation.

This method works by redirecting the user to the external payment processor's payment gateway, which then redirects to another URL on Convio to complete the transaction. For additional information on the information returned by a completed transaction, refer to the *donate* method.

Note: This API does not support a separate confirmation page on the calling site after the user enters their payment

information on PayPal, which is the default behavior when PayPal is configured as a payment gateway for Convio. You must contact Convio Support to request that this be changed, otherwise the label of the final button shown to the user in the PayPal flow will read "Continue" instead of "Pay Now," but clicking the button will complete the payment before returning to the calling code. The change will also affect the behavior of Convio donation forms that include PayPal as a payment method.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

billing.address.city

Required. The billing address of the credit card owner. If not supplied, defaults to donor.address.city.

Type xsd:string.

billing.address.state

Required. The billing address of the credit card owner. If not supplied, defaults to donor.address.state.

Type xsd:string.

billing.address.street1

Required. The billing address of the credit card owner. If not supplied, defaults to donor.address.street1.

Type xsd:string.

billing.address.zip

Required. The billing address of the credit card owner. If not supplied, defaults to donor.address.zip.

Type xsd:string.

billing.name.first

Required. The billing name of the credit card owner. If not supplied, defaults to donor.name.first.

Type xsd:string.

billing.name.last

Required. The billing name of the credit card owner. If not supplied, defaults to donor.name.last.

Type xsd:string.

donor.email

Required. The donor's email address.

Type xsd:string.

extproc

Required. The value designating which external payment processor to use.

Type xsd:string.

Options are:

- amazon
- paypal

form_id

Required. The ID of the donation form.

Type xsd:nonNegativeInteger.

level_id

Required. The ID of the donation level.

Type xsd:nonNegativeInteger.

ach_account_type

Optional. ACH account type (CHECKING/SAVINGS).

Type xsd:string.

Default is CHECKING.

Options are:
- CHECKING
- SAVINGS

ach_bank

Optional. ACH bank number (Canada).

Type xsd:string.

ach_transit

Optional. ACH transit number (Canada).

Type xsd:string.

additional_amount

Optional. Donation amount in addition to level amount.

Type xsd:string.

anonymous

Optional. Whether to mark the donation as "anonymous".

Type xsd:boolean.

billing.address.country

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.address.county

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.address.street2

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.address.street3

Optional. The billing address of the credit card owner.

Type xsd:string.

billing.name.middle

Optional. The billing name of the credit card owner.

Type xsd:string.

billing.name.profSuffix

Optional. The billing name of the credit card owner.

Type xsd:string.

billing.name.suffix

Optional. The billing name of the credit card owner.

Type xsd:string.

billing.name.title

Optional. The billing name of the credit card owner.

Type xsd:string.

designated.X.amount

Optional. The amount of a donation to designate to a designee. The "X" part of the parameter must be an integer used to correspond to designated.X.id or, in the case of write-in designations, designated.X.name and designated.X.contact. This allows multiple pairs of ID/amount for multiple designations.

Type xsd:nonNegativeInteger.

designated.X.id

Optional. The ID of a designee to which to designate all or part of a donation. The "X" part of the parameter must be an integer used to correspond to designated.X.amount. This allows multiple pairs of ID/amount for multiple designations.

Type xsd:nonNegativeInteger.

designated_write_in.X.amount

Optional. The amount of a donation to designate to a designee. The "X" part of the parameter must be an integer used to correspond to designated_write_in.X.name and designated_write_in.X.contact. This allows multiple pairs of ID/amount for multiple designations.

Type xsd:nonNegativeInteger.

designated_write_in.X.contact

Optional. The contact information of an organization to which to designate all or part of a donation. The "X" part of the parameter must be an integer used to correspond to designated_write_in.X.name and designated_write_in.X.amount. This allows multiple sets of name/contact/amount for multiple designations.

Type xsd:string.

designated_write_in.X.name

Optional. The name of an organization to which to designate all or part of a donation. The "X" part of the parameter must be an integer used to correspond to designated_write_in.X.contact and designated_write_in.X.amount. This allows multiple sets of name/contact/amount for multiple designations.

Type xsd:string.

donor.address.city

Optional. The donor's address.

Type xsd:string.

donor.address.country

Optional. The donor's address.

Type xsd:string.

donor.address.county

Optional. The donor's address.

Type xsd:string.

donor.address.state

Optional. The donor's address.

Type xsd:string.

donor.address.street1

Optional. The donor's address.

Type xsd:string.

donor.address.street2

Optional. The donor's address.

Type xsd:string.

donor.address.street3

Optional. The donor's address.

Type xsd:string.

donor.address.zip

Optional. The donor's address.

Type xsd:string.

donor.email_opt_in

Optional. The donor's email opt-in selection.

Type xsd:string.

donor.employer

Optional. The donor's employer.

Type xsd:string.

donor.name.first

Optional. The donor's name.

Type xsd:string.

donor.name.last

Optional. The donor's name.

Type xsd:string.

donor.name.middle

Optional. The donor's name.

Type xsd:string.

donor.name.profSuffix

Optional. The donor's name.

Type xsd:string.

donor.name.suffix

Optional. The donor's name.

Type xsd:string.

donor.name.title

Optional. The donor's name.

Type xsd:string.

donor.occupation

Optional. The donor's occupation.

Type xsd:string.

donor.phone

Optional. The donor's phone number.

Type xsd:string.

donor.phone_type

Optional. The type of the donor's phone number .

Type xsd:string.

Options are:

- home
- work
- other

ecard.copy_sender

Optional. Whether to send a copy of the e-card to the sender.

Type xsd:boolean.

ecard.id

Optional. The unique ID of the pre-defined Convio e-card template to use.

Type xsd:nonNegativeInteger.

ecard.message

Optional. The message body of the e-card.

Type xsd:string.

ecard.recipients

Optional. The email addresses of intended e-card recipients (RFC-5322 format) e.g. 'John Doe<user1@host.org>, Jane Roe<user2@host.com>'.

Type xsd:string.

ecard.send

Optional. Whether or not to send an e-card recognizing the gift.

Type xsd:boolean.

Default is false.

ecard.send_date

Optional. Date to send the e-card (ISO 8601 format) e.g. '2009-12-31'.

Type xsd:date.

ecard.subject

Optional. The subject of the e-card.

Type xsd:string.

finish_error_redirect

Optional. Specifies a URL that the method should redirect to on an error response from the external payment processor.

Type xsd:anyURI.

finish_success_redirect

Optional. Specifies a URL that the method should redirect to on a success response from the external payment processor.

Type xsd:anyURI.

fr_id

Optional. Specifies the ID of the TeamRaiser event associated with this donation.

Type xsd:nonNegativeInteger.

gift_aid

Optional. Gift aid status (UK Only).

Type xsd:string.

installment.duration

Optional. The duration of an installment plan donation, which is the number of payments that the donation is divided into. To make an installment plan donation, both installment.frequency and installment.duration must be specified.

Type xsd:nonNegativeInteger.

installment.frequency

Optional. The frequency of an installment plan donation, which is the interval between payments. To make an installment plan donation, both installment.frequency and installment.duration must be specified.

Type crm:donationSustainingFrequency.

Options are:
- one-time
- monthly
- quarterly
- semi-annually
- annually

joint_donor.name.first

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.last

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.middle

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.suffix

Optional. The name for a joint donor.

Type xsd:string.

joint_donor.name.title

Optional. The name for a joint donor.

Type xsd:string.

level_autorepeat

Optional. Whether or not the "recurring gift" option on the Donation Level shadow form should be selected. Valid only for Level shadow forms on which a recurring gift option has been defined.

Type xsd:boolean.

Default is false.

matching_eligible

Optional. Whether or not the donation is eligible for a matching gift.

Type xsd:boolean.

other_amount

Optional. User specified donation amount.

Type xsd:string.

premium_id

Optional. Unique ID of premium.

Type xsd:nonNegativeInteger.

premium_option

Optional. Premium option.

Type xsd:string.

remember_me

Optional. Whether to set a log-in cookie.

Type xsd:boolean.

send_receipt

Optional. Whether to send a receipt email to the donor. The default behavior is to send a receipt.

Type xsd:boolean.

Default is true.

send_registration_email

Optional. If a new constituent record is created for the donor, this specifies whether to send a registration "welcome" email to the donor. The default behavior is to send it.

Type xsd:boolean.

Default is true.

shipping.address.city

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.country

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.county

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.state

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.street1

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.street2

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.street3

Optional. Shipping address for premium.

Type xsd:string.

shipping.address.zip

Optional. Shipping address for premium.

Type xsd:string.

shipping.email

Optional. Email address of the recipient.

Type xsd:string.

shipping.name.first

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.last

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.middle

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.profSuffix

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.suffix

Optional. Premium recipient's name.

Type xsd:string.

shipping.name.title

Optional. Premium recipient's name.

Type xsd:string.

shipping.phone

Optional. Phone number of the recipient.

Type xsd:string.

soft_credit_id

Optional. Specifies the ID of the entity to which this donation will be soft credited. This will be either the event ID, team ID, or participant (constituent) ID, depending on the value specified for soft_credit_type.

Type xsd:nonNegativeInteger.

soft_credit_type

Optional. Specifies type of entity to which this donation will be soft credited. Donations that specify a soft credit type must also specify a soft_credit_id value.

Type xsd:string.

Options are:

- TR_PARTICIPANT
- TR_TEAM
- TR_EVENT
- TRIBUTE_GIFT

summary

Optional. The type of donation summary to return. There are three options:'data' returns a structured list of information about every form field; 'page' returns the HTML content (within the "body" element) of the "Thank you" page shown after donations via web page; 'both' returns both types of summary.

Type xsd:string.

Options are:

- data
- page
- both

sustaining.duration

Optional. The duration of a sustaining donation, which is the interval between payments. Specify zero to indicate that the donation should repeat indefinitely. To make a sustaining donation, both sustaining.frequency and sustaining.duration must be specified.

Type xsd:nonNegativeInteger.

sustaining.frequency

Optional. The frequency of a sustaining donation, which is how often the donation is repeated. To make a sustaining donation, both sustaining.frequency and sustaining.duration must be specified.

Type crm:donationSustainingFrequency.

Options are:

- one-time
- monthly
- quarterly
- semi-annually

- annually

teamraiser.message_to_participant

Optional. If this donation is a TeamRaiser Gift, the body of the message to send to the TeamRaiser participant.

Type xsd:string.

teamraiser.recognition_name

Optional. If this donation is a TeamRaiser Gift, the recognition name.

Type xsd:string.

teamraiser.show_gift_to_public

Optional. If this donation is a TeamRaiser Gift, whether or not to the TeamRaiser gift is publicly visible.

Type xsd:boolean.

tribute.honoree.deceased

Optional. Whether the tribute honoree is deceased. If set, it overrides "tribute.type" by setting it to "memorial" if true (deceased) and "tribute" if false (not deceased).

Type xsd:boolean.

tribute.honoree.name.first

Optional. The first name of the person to honor with the tribute. Takes precedence over tribute.honoree.name.full.

Type xsd:string.

tribute.honoree.name.full

Optional. The full name of the person to honor with the tribute. Only used if the first and last names are not specified.

Type xsd:string.

tribute.honoree.name.last

Optional. The last name of the person to honor with the tribute. Takes precedence over tribute.honoree.name.full.

Type xsd:string.

tribute.honoree.name.title

Optional. The name title of the person to honor with the tribute.

Type xsd:string.

tribute.message.body

Optional. The body of the tribute notification message.

Type xsd:string.

tribute.message.closing

Optional. The closing of the tribute notification message.

Type xsd:string.

tribute.message.include_amount

Optional. Whether to include the donation amount in the tribute notification message.

Type xsd:boolean.

tribute.message.signature

Optional. The signature of the tribute notification message.

Type xsd:string.

tribute.notify.address.city

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.country

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.county

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.state

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.street1

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.street2

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.street3

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.address.zip

Optional. The address of the person to notify of the tribute.

Type xsd:string.

tribute.notify.name.full

Optional. The full name of the person to notify of the tribute.

Type xsd:string.

tribute.notify.name.title

Optional. The name title of the person to notify of the tribute.

Type xsd:string.

tribute.type

Optional. The type of tribute.

Type xsd:string.

Options are:

- memorial

- tribute

validate

Optional. Whether to enforce validation of every field according to the specification in the ghost form ("true" or "false"). Some validation is always in effect, such as ensuring that the campaign, form, and level IDs reference published entities and that there is sufficient information to process the transaction.

Type xsd:boolean.

Default is false.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<donationResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <redirect>
    <url>https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_express-
checkout&amp;amp;token=EC-4PM50326PB7342636</url>
  </redirect>
</donationResponse>
```

JSON response

```
{"donationResponse":{"redirect":{"url":"https://www.sandbox.paypal.com/cgi-
bin/webscr?cmd=_express-checkout&amp;token=EC-4PM50326PB7342636"}}}
```

skipNextPayment

Skips the next payment for a recurring gift.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRRecurringAPI	SRRecurringAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRRecurringAPI?method=skipNextPayment &recurring_gift_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRRecurringAPI?method=skipNextPayment & recurring_gift_id =`*value* `& cons_id =`*value*

Usage Notes

This method causes the system to skip the next scheduled payment in a series of payments on a recurring gift.

You may use the following methods to allow a logged-in user to manage their recurring gifts themselves via the Client API or to manage recurring gifts as an adminstrator via the Server API:

- *calculateNextPaymentDate*
- *cancelRecurringGift*
- *getRecurringConfiguration*
- *getRecurringGiftDetails*
- *getRecurringGifts*
- *getRecurringHistory*
- *skipNextPayment*

- *updateRecurringBankAccountInfo*
- *updateRecurringCreditCardInfo*
- *updateRecurringGift_method*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

cons_id

Required. Specifies the ID of the constiuent.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<skipNextPaymentResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
</skipNextPaymentResponse>
```

JSON response

```
{"skipNextPaymentResponse":{"success":"true"}}
```

updateRecurringBankAccountInfo

Updates bank account ACH information for a recurring gift.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRRecurringAPI	SRRecurringAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRRecurringAPI?method=updateRecurringBankAccountInfo &recurring_gift_id=`*value* `&billing_city=`*value* `&billing_first_name=`*value* `&billing_last_name=`*value* `&billing_state=`*value* `&billing_street1=`*value* `&billing_zip=`*value* [`&ach_account_number=`*value*] [`&ach_account_type=`*value*] [`&ach_routing_number=`*value*] [`&ach_verify_number=`*value*] [`&billing_country=`*value*] [`&billing_middle_name=`*value*] [

`&billing_street2=`*value*] [`&billing_suffix=`*value*] [`&billing_title=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRRecurringAPI?method=updateRecurringBankAccountInfo & recurring_gift_id =`*value* `& billing_city =`*value* `& billing_first_name =`*value* `& billing_last_name =`*value* `& billing_state =`*value* `& billing_street1 =`*value* `& billing_zip =`*value* [`& ach_account_number =`*value*] [`& ach_account_type =`*value*] [`& ach_routing_number =`*value*] [`& ach_verify_number =`*value*] [`& billing_country =`*value*] [`& billing_middle_name =`*value*] [`& billing_street2 =`*value*] [`& billing_suffix =`*value*] [`& billing_title =`*value*] `& cons_id =`*value*

Usage Notes

This method allows a logged-in user to update their bank account information for a recurring gift.

You may use the following methods to allow a logged-in user to manage their recurring gifts themselves via the Client API or to manage recurring gifts as an adminstrator via the Server API:

- *calculateNextPaymentDate*
- *cancelRecurringGift*
- *getRecurringConfiguration*
- *getRecurringGiftDetails*
- *getRecurringGifts*
- *getRecurringHistory*
- *skipNextPayment*
- *updateRecurringBankAccountInfo*
- *updateRecurringCreditCardInfo*
- *updateRecurringGift_method*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

billing_city

Required. Specifies the billing city of a recurring gift.

Type xsd:string.

billing_first_name

Required. Specifies the billing first name of a recurring gift.

Type xsd:string.

billing_last_name

Required. Specifies the billing last name of a recurring gift.

Type xsd:string.

billing_state

Required. Specifies the billing state of a recurring gift.

Type xsd:string.

billing_street1

Required. Specifies the billing street1 of a recurring gift.

Type xsd:string.

billing_zip

Required. Specifies the billing zip of a recurring gift.

Type xsd:string.

ach_account_number

Optional. Specifies the ACH account number of a recurring gift.

Type xsd:string.

ach_account_type

Optional. Specifies the ACH account type of a recurring gift.

Type xsd:string.

ach_routing_number

Optional. Specifies the ACH routing number of a recurring gift.

Type xsd:string.

ach_verify_number

Optional. Specifies the ACH verify number of a recurring gift.

Type xsd:string.

billing_country

Optional. Specifies the billing country of a recurring gift.

Type xsd:string.

billing_middle_name

Optional. Specifies the billing first name of a recurring gift.

Type xsd:string.

billing_street2

Optional. Specifies the billing street2 of a recurring gift.

Type xsd:string.

billing_suffix

Optional. Specifies the billing suffix of a recurring gift.

Type xsd:string.

billing_title

Optional. Specifies the billing title of a recurring gift.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

billing_city

Required. Specifies the billing city of a recurring gift.

Type xsd:string.

billing_first_name

Required. Specifies the billing first name of a recurring gift.

Type xsd:string.

billing_last_name

Required. Specifies the billing last name of a recurring gift.

Type xsd:string.

billing_state

Required. Specifies the billing state of a recurring gift.

Type xsd:string.

billing_street1

Required. Specifies the billing street1 of a recurring gift.

Type xsd:string.

billing_zip

Required. Specifies the billing zip of a recurring gift.

Type xsd:string.

ach_account_number

Optional. Specifies the ACH account number of a recurring gift.

Type xsd:string.

ach_account_type

Optional. Specifies the ACH account type of a recurring gift.

Type xsd:string.

ach_routing_number

Optional. Specifies the ACH routing number of a recurring gift.

Type xsd:string.

ach_verify_number

Optional. Specifies the ACH verify number of a recurring gift.

Type xsd:string.

billing_country

Optional. Specifies the billing country of a recurring gift.

Type xsd:string.

billing_middle_name

Optional. Specifies the billing first name of a recurring gift.

Type xsd:string.

billing_street2

Optional. Specifies the billing street2 of a recurring gift.

Type xsd:string.

billing_suffix

Optional. Specifies the billing suffix of a recurring gift.

Type xsd:string.

billing_title

Optional. Specifies the billing title of a recurring gift.

Type xsd:string.

cons_id

Required. Specifies the ID of the constiuent.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateRecurringBankAccountInfoResponse xsi:schemaLocation="http://convio.com/
crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <updated>true</updated>
</updateRecurringBankAccountInfoResponse>
```

JSON response

```
{"updateRecurringBankAccountInfoResponse":{"updated":"true"}}
```

updateRecurringCreditCardInfo

Updates credit card information for a recurring gift.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRRecurringAPI	SRRecurringAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRRecurringAPI?method=updateRecurringCreditCardInfo &recurring_gift_id=`*value* `&billing_city=`*value* `&billing_first_name=`*value* `&billing_last_name=`*value* `&billing_state=`*value* `&billing_street1=`*value* `&billing_zip=`*value* [`&billing_country=`*value*] [`&billing_middle_name=`*value*] [`&billing_street2=`*value*] [`&billing_suffix=`*value*] [`&billing_title=`*value*] [`&credit_card_cvv=`*value*] [`&credit_card_exp_month=`*value*] [`&credit_card_exp_year=`*value*] [`&credit_card_number=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRRecurringAPI?method=updateRecurringCreditCardInfo &recurring_gift_id=`*value* `&billing_city=`*value* `&billing_first_name=`*value* `&billing_last_name=`*value* `&billing_state=`*value* `&billing_street1=`*value* `&billing_zip=`*value* [`&billing_country=`*value*] [`&billing_middle_name=`*value*] [`&billing_street2=`*value*] [`&billing_suffix=`*value*] [`&billing_title=`*value*] [`&credit_card_cvv=`*value*] [`&credit_card_exp_month=`*value*] [`&credit_card_exp_year=`*value*] [`&credit_card_number=`*value*] `&cons_id=`*value*

Usage Notes

This method allows a logged-in user to update their credit card information for a recurring gift.

Note: Because this method updates payment card information, for PCI-DSS compliance it must not be proxied.

You may use the following methods to allow a logged-in user to manage their recurring gifts themselves via the Client API or to manage recurring gifts as an adminstrator via the Server API:

- *calculateNextPaymentDate*
- *cancelRecurringGift*
- *getRecurringConfiguration*
- *getRecurringGiftDetails*
- *getRecurringGifts*
- *getRecurringHistory*
- *skipNextPayment*
- *updateRecurringBankAccountInfo*
- *updateRecurringCreditCardInfo*
- *updateRecurringGift_method*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

billing_city

Required. Specifies the billing city of a recurring gift.

Type xsd:string.

billing_first_name

Required. Specifies the billing first name of a recurring gift.

Type xsd:string.

billing_last_name

Required. Specifies the billing last name of a recurring gift.

Type xsd:string.

billing_state

Required. Specifies the billing state of a recurring gift.

Type xsd:string.

billing_street1

Required. Specifies the billing street1 of a recurring gift.

Type xsd:string.

billing_zip

Required. Specifies the billing zip of a recurring gift.

Type xsd:string.

billing_country

Optional. Specifies the billing country of a recurring gift.

Type xsd:string.

billing_middle_name

Optional. Specifies the billing first name of a recurring gift.

Type xsd:string.

billing_street2

Optional. Specifies the billing street2 of a recurring gift.

Type xsd:string.

billing_suffix

Optional. Specifies the billing suffix of a recurring gift.

Type xsd:string.

billing_title

Optional. Specifies the billing title of a recurring gift.

Type xsd:string.

credit_card_cvv

Optional. Specifies the credit card CVV of a recurring gift.

Type xsd:string.

credit_card_exp_month

Optional. Specifies the credit card expiration month of a recurring gift.

Type xsd:nonNegInteger.

credit_card_exp_year

Optional. Specifies the credit card expiration year of a recurring gift.

Type xsd:nonNegInteger.

credit_card_number

Optional. Specifies the credit card number of a recurring gift.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

billing_city

Required. Specifies the billing city of a recurring gift.

Type xsd:string.

billing_first_name

Required. Specifies the billing first name of a recurring gift.

Type xsd:string.

billing_last_name

Required. Specifies the billing last name of a recurring gift.

Type xsd:string.

billing_state

Required. Specifies the billing state of a recurring gift.

Type xsd:string.

billing_street1

Required. Specifies the billing street1 of a recurring gift.

Type xsd:string.

billing_zip

Required. Specifies the billing zip of a recurring gift.

Type xsd:string.

billing_country

Optional. Specifies the billing country of a recurring gift.

Type xsd:string.

billing_middle_name

Optional. Specifies the billing first name of a recurring gift.

Type xsd:string.

billing_street2

Optional. Specifies the billing street2 of a recurring gift.

Type xsd:string.

billing_suffix

Optional. Specifies the billing suffix of a recurring gift.

Type xsd:string.

billing_title

Optional. Specifies the billing title of a recurring gift.

Type xsd:string.

credit_card_cvv

Optional. Specifies the credit card CVV of a recurring gift.

Type xsd:string.

credit_card_exp_month

Optional. Specifies the credit card expiration month of a recurring gift.

Type xsd:nonNegInteger.

credit_card_exp_year

Optional. Specifies the credit card expiration year of a recurring gift.

Type xsd:nonNegInteger.

credit_card_number

Optional. Specifies the credit card number of a recurring gift.

Type xsd:string.

cons_id

Required. Specifies the ID of the constiuent.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateRecurringCreditCardInfoResponse xsi:schemaLocation="http://convio.com/
crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <updated>true</updated>
</updateRecurringCreditCardInfoResponse>
```

JSON response

```
{"updateRecurringCreditCardInfoResponse":{"updated":"true"}}
```

updateRecurringGift

Updates recurring gift information.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRRecurringAPI	SRRecurringAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRRecurringAPI?method=updateRecurringGift &recurring_gift_id=`*value* [`&amount=`*value*] `&next_payment_date=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRRecurringAPI?method=updateRecurringGift & recurring_gift_id =`*value* [`& amount =`*value*] `& next_payment_date =`*value* `& duration =`*value* `& cons_id =`*value*

Usage Notes

This method allows a logged-in user to update the next payment date, amount, or duration of a recurring gift.

You may use the following methods to allow a logged-in user to manage their recurring gifts themselves via the Client API or to manage recurring gifts as an adminstrator via the Server API:

- *calculateNextPaymentDate*
- *cancelRecurringGift*
- *getRecurringConfiguration*
- *getRecurringGiftDetails*
- *getRecurringGifts*
- *getRecurringHistory*
- *skipNextPayment*
- *updateRecurringBankAccountInfo*
- *updateRecurringCreditCardInfo*
- *updateRecurringGift_method*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

amount

Optional. Specifies the new amount of a recurring gift.

Type xsd:nonNegativeInteger.

next_payment_date

Required. Specifies the next payment date of a recurring gift. The date must be in the format MM-DD-YYYY

Type xsd:nonNegativeInteger.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

recurring_gift_id

Required. Specifies the ID of the recurring gift to retrieve.

Type xsd:nonNegativeInteger.

amount

Optional. Specifies the new amount of a recurring gift.

Type xsd:nonNegativeInteger.

next_payment_date

Required. Specifies the next payment date of a recurring gift. The date must be in the format MM-DD-YYYY

Type xsd:nonNegativeInteger.

duration

Required. Specifies the new duration of the recurring gift to retrieve. If forever is desired, 0 should be passed in, otherwise a positive integer.

Type xsd:nonNegativeInteger.

cons_id

Required. Specifies the ID of the constiuent.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateRecurringGiftResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
  <amountUpdated>true</amountUpdated>
  <amount>
    <decimal>1000</decimal>
    <formatted>$10.00</formatted>
  </amount>
  <nextPaymentDateUpdated>true</nextPaymentDateUpdated>
```

```
  <nextPaymentDate>2013-02-03-06:00</nextPaymentDate>
  <durationUpdated>true</durationUpdated>
  <duration>5</duration>
</updateRecurringGiftResponse>
```

JSON response

```
{"updateRecurringGiftResponse":{"amount":
{"formatted":"$10.00","decimal":"1000"},"duration":"5","durationUpdated":"true","amountUp
```

Group API

CRGroupAPI
`https://secure2.convio.net/organization/site/CRGroupAPI`
SRGroupAPI
`https://secure2.convio.net/organization/site/SRGroupAPI`

Description

The Group API provides methods to get information about user groups.

API Administrator accounts used for Server API calls to this API must belong to an Administrative Security Group that has "Use Convio Groups API" permission in the "Group Administration" permission type. The built-in "API Administrators" group has this permission by default.

getGroupInfo

Returns detailed information about a single user group.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRGroupAPI	SRGroupAPI	GET, POST	Yes.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRGroupAPI?method=getGroupInfo &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&group_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]
Server API Syntax
`https://`*secure2.convio.net/organization*`/site/SRGroupAPI?method=getGroupInfo & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& group_id =`*value*

Usage Notes

This method requires a valid group_id parameter and returns detailed information about the specified group.

Client API callers must be logged in and must be administrator users with appropriate Group Management rights. Server API callers must have "Group Management/Use Convio Groups API" permission.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

group_id

Required. The unique identifier of the group to retrieve.

Type xsd:nonNegativeInteger.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

group_id

Required. The unique identifier of the group to retrieve.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getGroupInfoResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <groupInfo>
    <id>0</id>
    <name>Sample Group 1 Name</name>
    <description>Sample Group 1 Description</description>
    <createTimestamp>2013-02-03T05:18:23.791-06:00</createTimestamp>
    <modifyTimestamp>2013-02-03T05:18:23.791-06:00</modifyTimestamp>
    <lastBuildTimestamp>2013-02-03T05:18:23.791-06:00</lastBuildTimestamp>
    <statsUpdateTimestamp>2013-02-03T05:18:23.791-06:00</statsUpdateTimestamp>
    <groupMode>STATIC_REBUILDABLE</groupMode>
    <securityMode>ADMINISTRATOR</securityMode>
    <numMembers>2831</numMembers>
    <numActiveMembers>2279</numActiveMembers>
    <emailStats>
      <numHardBounce>0</numHardBounce>
      <numSoftBounce>0</numSoftBounce>
      <numOptedOut>0</numOptedOut>
      <numSendable>0</numSendable>
    </emailStats>
    <securityCategory>
      <id>1</id>
      <label>Sample Security Category</label>
    </securityCategory>
    <groupType>
      <id>334</id>
      <name>Sample Group Type</name>
    </groupType>
  </groupInfo>
</getGroupInfoResponse>
```

JSON response

```
{"getGroupInfoResponse":{"groupInfo":{"groupType":
{"id":"334","name":"Sample Group Type"},"emailStats":
{"numOptedOut":"0","numSendable":"0","numSoftBounce":"0","numHardBounce":"0"},"lastBuildT
{"id":"1","label":"Sample Security
 Category"},"id":"0","createTimestamp":"2013-02-03T05:18:23.800-06:00","statsUpdateTimest
 Group 1 Description","numActiveMembers":"2279","name":"Sample Group 1
 Name","numMembers":"2831","securityMode":"ADMINISTRATOR"}}}
```

getGroupMembers

This method returns the list of the constituents in a group. By default, each group member's constituent ID, member ID, first and last names, primary email address and home address are returned.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=getGroupMembers`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&group_id=`*value* [`&fields=`*value*] [`&max_number=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=getGroupMembers &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& group_id =`*value* [`& fields =`*value*] [`& max_number =`*value*]

Usage Notes

This method returns personal information for members of a user group. It is only available for administrative users having the appropriate permission to review member lists. If this method is called as a Client API, the administrator must be logged in and an authentication token must be provided in the request.

By default, the method returns constituent ID, member ID, name, primary mailing address and primary email address. The fields parameter can be used to specify a comma-separated list of other fields from the constituent record.The *listUserFields* method returns the list of available fields, which may differ from one Convio site to another.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

group_id

Required. The Group ID key referencing the Group record.

Type xsd:nonNegativeInteger.

fields

Optional. A comma-separated list of field names of constituent record fields to return for each member.

Type xsd:string.

Default is cons_id, member_id, first_name, last_name, primary_email, home_street1, home_street2, home_street3, home_city, home_stateprov, home_county, home_zip.

max_number

Optional. The maximum number of members to return (up to 1000).

Type xsd:nonNegativeInteger.

Default is 1000.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

group_id

Required. The Group ID key referencing the Group record.

Type xsd:nonNegativeInteger.

fields

Optional. A comma-separated list of field names of constituent record fields to return for each member.

Type xsd:string.

Default is cons_id, member_id, first_name, last_name, primary_email, home_street1, home_street2, home_street3, home_city, home_stateprov, home_county, home_zip.

max_number

Optional. The maximum number of members to return (up to 1000).

Type xsd:nonNegativeInteger.

Default is 1000.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getGroupMembersResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <total_number>2</total_number>
  <number_returned>2</number_returned>
  <member>
    <cons_id>1000121</cons_id>
    <member_id>Member ID</member_id>
    <name>
      <first>First</first>
      <last>Last</last>
    </name>
    <email>
      <primary_address>email@address.net</primary_address>
    </email>
    <primary_address>
      <street1>Street1</street1>
      <street2>Street2</street2>
      <street3>Street3</street3>
      <city>City</city>
      <state>State</state>
      <zip>Zip</zip>
      <county>County</county>
    </primary_address>
  </member>
</getGroupMembersResponse>
```

JSON response

```
{"getGroupMembersResponse":{"member":{"primary_address":
{"street2":"Street2","zip":"Zip","street1":"Street1","county":"County","state":"State","s
{"primary_address":"email@address.net"},"name":
{"last":"Last","first":"First"},"cons_id":"1000121","member_id":"Member
 ID"},"number_returned":"2","total_number":"2"}}
```

listGroups

Returns a list of user groups

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRGroupAPI	SRGroupAPI	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRGroupAPI?method=listGroups`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&allow_cross_center=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&selection_mode=`*EMAIL* | *ADMIN_SECURITY* | *MEMBERSHIP* | *HEIRARCHY* | *NOT_BUILT_IN* | *REPORTWRITER_SHAREES* | *STATES*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRGroupAPI?method=listGroups &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] [`& allow_cross_center =`*value*] [`& list_filter_text =`*value*] [`& list_page_offset =`*value*] [`& list_page_size =`*value*] [`& list_record_offset =`*value*] [`& selection_mode =`*EMAIL* | *ADMIN_SECURITY* | *MEMBERSHIP* | *HEIRARCHY* | *NOT_BUILT_IN* | *REPORTWRITER_SHAREES* | *STATES*]

Usage Notes

This method returns a list of the Groups defined for a particular Convio site. It supports basic list paging and sort order criteria. The list can be filtered based on Group selection mode.

Client API callers must be logged in and must be administrator users with appropriate Group Management rights. Server API callers must have "Group Management/Use Convio Groups API" permission.

The MEMBERSHIP option for selection_mode does not return interests groups. To obtain a list of the constituent's interests, use instead *listInterests*.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

allow_cross_center

Optional. Indicates groups results may be cross centers in single center mode.

Type xsd:boolean.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

selection_mode

Optional. Specifies a context for the results, which can differ according to how they are expected to be used

Type xsd:token.

Default is NOT_BUILT_IN.

Options are:

- EMAIL
- ADMIN_SECURITY
- MEMBERSHIP
- HEIRARCHY
- NOT_BUILT_IN
- REPORTWRITER_SHAREES
- STATES

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

allow_cross_center

Optional. Indicates groups results may be cross centers in single center mode.

Type xsd:boolean.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

selection_mode

Optional. Specifies a context for the results, which can differ according to how they are expected to be used

Type xsd:token.

Default is NOT_BUILT_IN.

Options are:

- EMAIL
- ADMIN_SECURITY
- MEMBERSHIP
- HEIRARCHY
- NOT_BUILT_IN
- REPORTWRITER_SHAREES
- STATES

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listGroupsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <groupInfo>
    <id>0</id>
    <name>Sample Group 1 Name</name>
    <description>Sample Group 1 Description</description>
    <statsUpdateTimestamp>2013-02-03T05:18:23.886-06:00</statsUpdateTimestamp>
    <numMembers>2831</numMembers>
    <numActiveMembers>2279</numActiveMembers>
    <groupMode>STATIC_REBUILDABLE</groupMode>
    <securityMode>ADMINISTRATOR</securityMode>
  </groupInfo>
</listGroupsResponse>
```

JSON response

```
{"listGroupsResponse":{"groupInfo":
{"id":"0","statsUpdateTimestamp":"2013-02-03T05:18:23.887-06:00","groupMode":"STATIC_REBU
 Group 1 Description","numMembers":"2831","name":"Sample Group 1
 Name","securityMode":"ADMINISTRATOR"}}}
```

Calendar Event API

CROrgEventAPI
`https://secure2.convio.net/organization/site/CROrgEventAPI`

Description

The Calendar Event API allows you to retrieve Event information from the Convio web service and re-display the Event details in a third-party web site or web-based application. They may be used to populate a custom calendar, or to port event information to third-party sites such as Eventful or Upcoming.org.

The Calendar Event API supports either HTTP GET or POST requests. These APIs are client-only. These APIs do not require authentication; they are called in the context of an anonymous user.

The API communicates with the Convio Calendar Events database and provides the following methods:

getChildEvents

This method retrieves the active child events, regardless of security category, associated with a specified parent event. Only active parent events with an event type of 3 (multi-part event) may contain child events. Please note that the eventUrl is not returned for child events whereas it is for other events.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CROrgEventAPI	SROrgEventAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CROrgEventAPI?method=getChildEvents`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&event_id=`*value* [`&include_private_events=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SROrgEventAPI?method=getChildEvents`
`&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] `&event_id=`*value* [`&include_private_events=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

event_id

Required. Specifies the ID of the "parent" event for which to retrieve a list of "child" events.

Type xsd:nonNegativeInteger.

include_private_events

Optional. Specifies whether private events will be included in the results

Type xsd:boolean.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

event_id

Required. Specifies the ID of the "parent" event for which to retrieve a list of "child" events.

Type xsd:nonNegativeInteger.

include_private_events

Optional. Specifies whether private events will be included in the results

Type xsd:boolean.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getChildEventsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <event>
    <name>Event Name</name>
    <id>1000</id>
    <description>Event Description</description>
    <eventDate>
      <startDate>2013-02-03-06:00</startDate>
      <endDate>2013-02-03-06:00</endDate>
      <startTime>05:18:27.965-06:00</startTime>
      <endTime>05:18:27.965-06:00</endTime>
    </eventDate>
    <eventUrl>http://www.event.org</eventUrl>
    <eventType>1</eventType>
    <eventLocation>
      <city>City</city>
      <street1>Street1</street1>
      <stateProv>State</stateProv>
      <zip>ZIP</zip>
      <street2>Street2</street2>
    </eventLocation>
  </event>
</getChildEventsResponse>
```

JSON response

```
{"getChildEventsResponse":{"event":{"id":"1000","eventLocation":
{"street2":"Street2","stateProv":"State","zip":"ZIP","street1":"Street1","city":"City"},"
{"startTime":"05:18:27.965-06:00","startDate":"2013-02-03-06:00","endDate":"2013-02-03-06
 Description","name":"Event Name","eventUrl":"http://
www.event.org","eventType":"1"}}}
```

getDayEvents

This method retrieves the active events, regardless of security category, occurring on a specified day.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CROrgEventAPI	SROrgEventAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CROrgEventAPI?method=getDayEvents`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&day=`*value* `&month=`*value* `&year=`*value* [`&include_private_events=`*value*] [`&interests=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SROrgEventAPI?method=getDayEvents &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& day =`*value* `& month =`*value* `& year =`*value* [`& include_private_events =`*value*] [`& interests =`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

day

Required. Specifies the day to retrieve (e.g. 1-31).

Type xsd:nonNegativeInteger.

month

Required. Specifies the month to retrieve (e.g. 1-12).

Type xsd:nonNegativeInteger.

year

Required. Specifies the month to retrieve (e.g. 2009).

Type xsd:nonNegativeInteger.

include_private_events

Optional. Specifies whether private events will be included in the results

Type xsd:boolean.

interests

Optional. Specifies a comma-separated list of interest names or IDs by which to filter the results. Names and IDs can be mixed in the same list. Items in the list that are all digits are assumed to be IDs. IDs or names that do match available interests are ignored without error. Spaces between items are ignored; spaces within a name are preserved.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

day

Required. Specifies the day to retrieve (e.g. 1-31).

Type xsd:nonNegativeInteger.

month

Required. Specifies the month to retrieve (e.g. 1-12).

Type xsd:nonNegativeInteger.

year

Required. Specifies the month to retrieve (e.g. 2009).

Type xsd:nonNegativeInteger.

include_private_events

Optional. Specifies whether private events will be included in the results

Type xsd:boolean.

interests

Optional. Specifies a comma-separated list of interest names or IDs by which to filter the results. Names and IDs can be mixed in the same list. Items in the list that are all digits are assumed to be IDs. IDs or names that do match available interests are ignored without error. Spaces between items are ignored; spaces within a name are preserved.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getDayEventsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <event>
    <name>Event Name</name>
    <id>1000</id>
    <description>Event Description</description>
    <eventDate>
      <startDate>2013-02-03-06:00</startDate>
      <endDate>2013-02-03-06:00</endDate>
      <startTime>05:18:27.991-06:00</startTime>
      <endTime>05:18:27.991-06:00</endTime>
    </eventDate>
    <eventUrl>http://www.event.org</eventUrl>
    <eventType>1</eventType>
    <eventLocation>
      <city>City</city>
      <street1>Street1</street1>
      <stateProv>State</stateProv>
      <zip>ZIP</zip>
      <street2>Street2</street2>
    </eventLocation>
  </event>
</getDayEventsResponse>
```

JSON response

```
{"getDayEventsResponse":{"event":{"id":"1000","eventLocation":
{"street2":"Street2","stateProv":"State","zip":"ZIP","street1":"Street1","city":"City"},'
{"startTime":"05:18:27.992-06:00","startDate":"2013-02-03-06:00","endDate":"2013-02-03-06
 Description","name":"Event Name","eventUrl":"http://
www.event.org","eventType":"1"}}}
```

getEvent

This method retrieves the individual event record for a given event ID.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CROrgEventAPI	SROrgEventAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CROrgEventAPI?method=getEvent`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&event_id=`*value*

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SROrgEventAPI?method=getEvent &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& event_id =`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

event_id

Required. Specifies the ID of the event to retrieve.

Type xsd:nonNegativeInteger.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

event_id

Required. Specifies the ID of the event to retrieve.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getEventResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <event>
    <name>Event Name</name>
    <id>1000</id>
    <description>Event Description</description>
    <eventDate>
      <startDate>2013-02-03-06:00</startDate>
      <endDate>2013-02-03-06:00</endDate>
      <startTime>05:18:27.951-06:00</startTime>
      <endTime>05:18:27.951-06:00</endTime>
    </eventDate>
    <eventUrl>http://www.event.org</eventUrl>
    <eventType>1</eventType>
    <eventLocation>
      <city>City</city>
      <street1>Street1</street1>
      <stateProv>State</stateProv>
      <zip>ZIP</zip>
      <street2>Street2</street2>
    </eventLocation>
  </event>
</getEventResponse>
```

JSON response

```
{"getEventResponse":{"event":{"id":"1000","eventLocation":
{"street2":"Street2","stateProv":"State","zip":"ZIP","street1":"Street1","city":"City"},"
{"startTime":"05:18:27.953-06:00","startDate":"2013-02-03-06:00","endDate":"2013-02-03-06
 Description","name":"Event Name","eventUrl":"http://
www.event.org","eventType":"1"}}}
```

getMonthEvents

This method retrieves the active events, regardless of security category, occurring within a specified month and year.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CROrgEventAPI	SROrgEventAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CROrgEventAPI?method=getMonthEvents &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&month=`*value* `&year=`*value* [`&include_private_events=`*value*] [`&interests=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SROrgEventAPI?method=getMonthEvents & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& month =`*value* `& year =`*value* [`& include_private_events =`*value*] [`& interests =`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

month

Required. Specifies the month to retrieve (e.g. 1-12).

Type xsd:nonNegativeInteger.

year

Required. Specifies the month to retrieve (e.g. 2009).

Type xsd:nonNegativeInteger.

include_private_events

Optional. Specifies whether private events will be included in the results

Type xsd:boolean.

interests

Optional. Specifies a comma-separated list of interest names or IDs by which to filter the results. Names and IDs can be mixed in the same list. Items in the list that are all digits are assumed to be IDs. IDs or names that do match available interests are ignored without error. Spaces between items are ignored; spaces within a name are preserved.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

month

Required. Specifies the month to retrieve (e.g. 1-12).

Type xsd:nonNegativeInteger.

year

Required. Specifies the month to retrieve (e.g. 2009).

Type xsd:nonNegativeInteger.

include_private_events

Optional. Specifies whether private events will be included in the results

Type xsd:boolean.

interests

Optional. Specifies a comma-separated list of interest names or IDs by which to filter the results. Names and IDs can be mixed in the same list. Items in the list that are all digits are assumed to be IDs. IDs or names that do match available interests are ignored without error. Spaces between items are ignored; spaces within a name are preserved.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getMonthEventsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <event>
    <name>Event Name</name>
    <id>1000</id>
    <description>Event Description</description>
    <eventDate>
      <startDate>2013-02-03-06:00</startDate>
      <endDate>2013-02-03-06:00</endDate>
      <startTime>05:18:27.977-06:00</startTime>
      <endTime>05:18:27.977-06:00</endTime>
    </eventDate>
    <eventUrl>http://www.event.org</eventUrl>
    <eventType>1</eventType>
    <eventLocation>
      <city>City</city>
      <street1>Street1</street1>
      <stateProv>State</stateProv>
      <zip>ZIP</zip>
      <street2>Street2</street2>
    </eventLocation>
  </event>
</getMonthEventsResponse>
```

JSON response

```
{"getMonthEventsResponse":{"event":{"id":"1000","eventLocation":
{"street2":"Street2","stateProv":"State","zip":"ZIP","street1":"Street1","city":"City"},"
{"startTime":"05:18:27.978-06:00","startDate":"2013-02-03-06:00","endDate":"2013-02-03-06
 Description","name":"Event Name","eventUrl":"http://
www.event.org","eventType":"1"}}}
```

Single Sign-on API

CRConsAPI
`https://secure2.convio.net/organization/site/CRConsAPI`
SRConsAPI
`https://secure2.convio.net/organization/site/SRConsAPI`

Description

The Single Sign-on API is implemented by the Constituent API servlets (CRConsAPI and SRConsAPI). It provides cross-domain authentication services using either Convio or a trusted host as the master authentication provider. These APIs must be used obtain an authentication token for any client API call that requires authentication, i.e. where the call is made on behalf of a logged-in user.

The `getSingleSignOnToken` and `singleSignOn` API methods are used to log in a user and establish a session when the caller assumes responsibility for authenticating the user; see *Single Sign-on (Convio as client)*.

The `login`, `loginTest`, and `authenticateUser` API methods are used to log in a user and establish a session using Convio as the master authentication server; see *Single Sign-on (Convio as master)*.

This API also provides methods to end a session (`logoff`) and allow a logged-in user or an administrator to change passwords (`changePassword`).

The Single Sign-on API exposes the following methods:

authenticateUser

This method is similar to the login method. It can be used to authenticate and establish a session for a user who is in the Convio constituent database, but who has not logged in to the site for the first time. This method takes an organization-specific combination of attributes (such as membership number and zip code) to authenticate the user and establish a session.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Server.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=authenticateUser`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
Server API Syntax
`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=authenticateUser &`
`api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*]

Usage Notes

The `authenticateUser` method is similar to the login method. It can be used to log in a user and establish a session with the Convio server using an alternative means of authentication, such as a name and membership ID number, if that user has never logged in before and has not established a user name and password on the system.

To use this method, you must first define which columns in your constituent database to use for authentication. Under Setup/Database Configuration, check "Use for Authentication" next to the columns to use, and note the Column Names, which you will use as parameters when calling this method. If no fields in the database are marked "Use for Authentication," this method returns the following error: "No fields are configured to support authentication."

You may not have access to modify your site's Database Configuration. If not, you must open a case with Convio Support specifying which Constituent profile fields you want flagged "Use for Authentication," and requesting the column names of those fields to use in the API call.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<loginResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cons_id>1001002</cons_id>
  <timestamp>1359890307</timestamp>
  <signature>4c22627363a21f889d084ee12f608aa9</signature>
  <token xsi:nil="true"/>
</loginResponse>
```

JSON response

```
{"loginResponse":
{"timestamp":"1359890307","token":null,"cons_id":"1001002","signature":"4c22627363a21f889
```

Examples

A user may be recorded in the database without ever having logged in to the system. They have a membership card that uniquely identifies the account information and that is considered relatively secure. In actual usage, it is a best practice to require at least 2 pieces of information from the user to authenticate (e.g. last name and membership number).

The solution in this scenario is to design a secure form that uses POST to the authenticateUser method at this URL:

```
https://securex.convio.net/[org]/site/CRConsAPI
```

The form should look something like this:

```
<form method="post" action="https://securex.convio.net/yoursite/site/
CRConsAPI">
  <input type="hidden" name="api_key" id="api_key" value="open" />
  <input type="hidden" name="v" id="v" value="1.0" />
  <input type="hidden" name="method" id="method" value="authenticateUser" />
  <input type="hidden" name="success_redirect" value="http://
yourothersite.org/login_page.html?cons_id=${loginResponse/cons_id}" >
  <input type="hidden" name="error_redirect" value="http://yourothersite.org/
login_page.html?code=${errorResponse/code}&message=${errorResponse/message}" >
  <input type="hidden" name="sign_redirects" id="sign_redirects"
 value="true" />
  <table>
  <tr>
    <td>Last Name:</td>
    <td><input name="last_name" type="text" size="15" maxlength="100"></td>
  <tr>
    <td>Membership Number:</td>
    <td><input name="member_id" type="text" size="15" maxlength="100"></td>
  </tr>
  </table>
```

```
  <input type="submit" value="Submit" name="Submit" />
</form>
```

The Input fields displayed to the user will be based on site-specific configuration in the Constituent Database (used for Authentication flag). Typically, this configuration should be done by Convio Support.

changePassword

Used to change a user's password in the Convio constituent database.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=changePassword` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&old_password=`*value* `&user_password=`*value* `&retype_password=`*value* `&reminder_hint=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=changePassword &` `api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& old_password =`*value* `& user_password =`*value* `& retype_password =`*value* `& reminder_hint =`*value* [`& cons_id =`*value*] [`& member_id =`*value*] [`& primary_email =`*value*]

Usage Notes

If called as a Client API, this method requires authentication. If the caller is an administrator (normally only the case using the Server API):

1. One of cons_id, member_id, or primary_email must be specified in order to locate the record of the constituent whose password is to be changed.
2. The order of precedence of the search terms is (a) cons_id (b) member_id, then (c) primary_email. If a record is not found or if multiple records are returned by any specified search term evaluated in this order, the method returns an error.

If the requester is a constituent using the Client API:

1. An authentication token representing the logged-in user must be passed.
2. Only the password of the logged in user may be changed.

Note: For security reasons, you cannot use this API to change an administrator's password.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

old_password

Required. The user's existing password.

Type xsd:string.

user_password

Required. The new password.

Type xsd:string.

retype_password

Required. The new password re-typed by the user to confirm.

Type xsd:string.

reminder_hint

Required. Password hint entered by the user.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

old_password

Required. The user's existing password.

Type xsd:string.

user_password

Required. The new password.

Type xsd:string.

retype_password

Required. The new password re-typed by the user to confirm.

Type xsd:string.

reminder_hint

Required. Password hint entered by the user.

Type xsd:string.

cons_id

Optional. The constituent ID (Convio ID) that uniquely identifies the constituent. This value is automatically assigned when a new constituent record is created. It can be used to uniquely identify a constituent record to read or update it. Only available when called using an administrator account.

Type xsd:string.

member_id

Optional. The member ID (Client ID) that uniquely identifies the constituent.

Type xsd:string.

primary_email

Optional. The primary email address that uniquely identifies the constituent.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateConsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cons_id>1000121</cons_id>
```

```
  <message>User updated.</message>
</updateConsResponse>
```

JSON response

```
{"updateConsResponse":{"message":"User updated.","cons_id":"1000121"}}
```

getSingleSignOnToken

This server-only method returns a single sign-on token that can be used in client-side requests to automatically log a user into the system. This is to be used in the case where a web site external to Convio has authenticated the site visitor and wants to direct the user's browser to the Convio system and have them automatically logged in to the Convio system. The external web server, after verifying the identity of the visitor, calls this API method to get a token that will log the user into the Convio system when sent by the visitor's browser in a subsequent HTTP request. This token is only valid for the specific user and only for a limited time (typically, a few minutes). To log in to the Convio system and visit most pages, the token need only be presented one time to initiate the logged-in session. However, if other API methods are used during the visit, this token should be included in all API requests to verify that the API methods are being called from a trusted source. While the token cannot be used after its time limit to log a user in, it can be used for the duration of the current session to verify the trustworthiness of the source.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRConsAPI	POST	Yes.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=getSingleSignOnToken` `&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] [`&cons_id=`*value*] [`&member_id=`*value*]

Usage Notes

This **server-only** method returns a single sign-on token that can be passed in a client-side requests by *singlSignOn* to log a user authenticated by a third-party server into the Convio system. It is to be used in the case where a web site external to Convio has authenticated the site visitor and wants to direct the user's browser to the Convio system and have them automatically logged in to the Convio system, or access Convio client APIs in the context of an authenticated user session.

Note: Use this API only if the third-party server will authenticate the user. To use the Convio server as the master authenticator, use *login* or *authenticateUser* instead.

This method must be invoked over a secure channel by a host system that is recognized by the Convio server in its approved IP White List, and must stipulate a valid cons_id or member_id of a constituent in the Convio database.

If successful, this method returns a time-sensitive token to the caller. This token must be passed from the user's browser in a call to `getSingleSignOn` to establish an authenticated user session. It can then be passed in the sso_auth_token parameter of client API calls that require authentication.

Note: The token returned must be used within 15 minutes to initiate a logged-in session from the client browser using the singleSignOn client API. It can then be passed to other client APIs requiring authentication while the session remains active.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Optional. The Convio constituent ID for the user on whose behalf the token is requested. Optional, however, either a unique cons_id or a unique member_id parameter must be supplied in order for the call to succeed.

Type xsd:nonNegativeInteger.

member_id

Optional. The client's unique member ID , as recorded in the Convio constituent database, for the user on whose behalf the token is requested. Optional, however, either a unique cons_id or a unique member_id parameter must be supplied in order for the call to succeed.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getSingleSignOnTokenResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

 <token>5F8p37tMzmB9qfjoXd9i2nkb_INrxXfa8wL7rbpSQkm5gwEgRQV5F6YmJwfKbIIgbHbeqaJcGNLWGdJq
token>
  <cons_id>1000543</cons_id>
</getSingleSignOnTokenResponse>
```

JSON response

```
{"getSingleSignOnTokenResponse":
{"token":"5F8p37tMzmB9qfjoXd9i2nkb_INrxXfa8wL7rbpSQkm5gwEgRQV5F6YmJwfKbIIgbHbeqaJcGNLWGd
```

login

Use the login method to log an existing user in to the Convio-powered site based on a user_name and password or to send those credentials to the user by email.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	SRConsAPI	POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=login &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&email=`*value*] [`&logout=`*value*] [`&password=`*value*] [`&remember_me=`*value*] [`&send_user_name=`*value*] [`&user_name=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRConsAPI?method=login & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`&`

`suppress_response_codes` =*value*] [& `email` =*value*] [& `logout` =*value*] [& `password` =*value*] [& `remember_me` =*value*] [& `send_user_name` =*value*] [& `user_name` =*value*]

Usage Notes

The `login` method permits users to log in to the Convio system and establish a session. If the user successfully logs in, this method returns the user's Constituent ID and an authorization token that can be used access other APIs that require authentication during the logged-in user's current session.

This API is generally invoked by sending a redirect from server code or in an onload JavaScript action.

Even though this API is actually invoked over a secure URL, you should invoke the API using the insecure domain for the client and allow the Convio application to redirect the API call to the secure domain. This will ensure that if the user has only established a session on the insecure domain that it will be checked first. It also ensures that the API checks for a login cookie on both the insecure and secure domains before providing the response. Failure to follow this convention will result in the API erroneously flagging users as not being logged in.

When the `login` method is successfully called from a client using the Client API it will log in the session that processes the request for that user. It will also attempt to push a session cookie to the user's browser. When called with the redirect or success_redirect parameters, it may process an intermediate redirect in order to ensure that session cookies are pushed for both the secure and insecure domains.

When the `login` method is successfully called from a server using the Server API, it will not change the state of the session that processes the request nor will it attempt to push session cookies.

The send_user_name and email parameters may be used to retrieve a forgotten user name and password for the user and send them to a previously-registered email address.

See also *authenticateUser*, *loginTest*, and *logout*.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

email

Optional. Email address to which to send user name and password if 'send_user_name=true'

Type xsd:string.

logout

Optional. If 'true,' closes the current session and logs the user out. Equivalent to the logout method.

Type xsd:boolean.

Default is false.

password

Optional. Password of the constituent to log in.

Type xsd:string.

remember_me

Optional. If set, a log-in cookie is pushed to the browser to recognize the current user on his next visit.

Type xsd:boolean.

Default is false.

send_user_name

Optional. If 'true', sends the user's user name and password to the address specified in the 'email' parameter if the email address matches one that is registered for the user.

Type xsd:boolean.

Default is false.

user_name

Optional. User name of the constituent to log in.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

email

Optional. Email address to which to send user name and password if 'send_user_name=true'

Type xsd:string.

logout

Optional. If 'true,' closes the current session and logs the user out. Equivalent to the logout method.

Type xsd:boolean.

Default is false.

password

Optional. Password of the constituent to log in.

Type xsd:string.

remember_me

Optional. If set, a log-in cookie is pushed to the browser to recognize the current user on his next visit.

Type xsd:boolean.

Default is false.

send_user_name

Optional. If 'true', sends the user's user name and password to the address specified in the 'email' parameter if the email address matches one that is registered for the user.

Type xsd:boolean.

Default is false.

user_name

Optional. User name of the constituent to log in.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<loginResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cons_id>1001002</cons_id>
  <timestamp>1359890307</timestamp>
  <signature>4c22627363a21f889d084ee12f608aa9</signature>
  <token xsi:nil="true"/>
</loginResponse>
```

JSON response

```
{"loginResponse":
{"timestamp":"1359890307","token":null,"cons_id":"1001002","signature":"4c22627363a21f88
```

Examples

Logging In

When a user enters a user name and password on a partner site's login page, the partner wants to authenticate with Convio and redirect that user to a content page if successful, or back to the login page if the authentication was unsuccessful.

The solution in this scenario is to design a secure form that uses POST to the login method at this URL:

```
    https://securex.convio.net/[org]/site/CRConsAPI
```

The form should look something like this:

```
<form method="post" action="https://securex.convio.net/yoursite/site/
CRConsAPI">
  <input type="hidden" name="api_key" id="api_key" value="open" />
  <input type="hidden" name="v" id="v" value="1.0" />
  <input type="hidden" name="method" id="method" value="login" />
  <input type="hidden" name="success_redirect" value="http://
yourothersite.org/login_page.html?cons_id=${loginResponse/cons_id}" >
  <input type="hidden" name="error_redirect" value="http://yourothersite.org/
login_page.html?code=${errorResponse/code}&message=${errorResponse/message}" >
  <input type="hidden" name="sign_redirects" id="sign_redirects"
 value="true" />
  <table>
  <tr>
    <td>User Name:</td>
    <td><input name=user_name type=text size=15 maxlength="100"></td>
  </tr>
  <tr>
  <td>Password:</td>
    <td><input name=password type=password size=15 maxlength="100"></td>
  </tr>
  <tr>
  <td>Remember Me:</td>
    <td><input name=remember_me type=checkbox></td>
  </tr>
  </table>
  <input type="submit" value="Submit" name="Submit" />
</form>
```

When the form is submitted, the data will be posted to the API. In the event of a correct username/password combination, a session will be established on Convio and the user's browser will be redirected back to the partner system. The redirect will contain the cons_id of the user and the signature. The partner system should then log that user in and display the content provided that the user is authorized. In the event of a login failure, the user's browser will be redirected back to the login page with an error message in the URL. The login page should then take that error message from the URL and display it on the page to the user.

Retrieving a Forgotten User Name or Password

When a user has lost or forgotten their user name and password, they expect to be provided some means to retrieve that information from the system, typically through email.

The solution in this scenario is to design a secure form that uses POST to the login method with the option send_user_name=true at this URL:

```
    https://securex.convio.net/[org]/site/CRConsAPI
```

The form should look something like this:

```
<form method="post" action="https://securex.convio.net/yoursite/site/
CRConsAPI">
  <input type="hidden" name="api_key" id="api_key" value="open" />
  <input type="hidden" name="v" id="v" value="1.0" />
  <input type="hidden" name="method" id="method" value="login" />
  <input type="hidden" name="send_user_name" id="send_user_name"
 value="true" />
  <input type="hidden" name="success_redirect" value="http://
yourothersite.org/login_page.html?user_name_sent=true" >
  <input type="hidden" name="error_redirect" value="http://yourothersite.org/
login_page.html?code=${errorResponse/code}&message=${errorResponse/message}" >
  <input type="hidden" name="sign_redirects" id="sign_redirects"
 value="true" />
  <table>
  <tr>
    <td>Email you registered with:</td>
    <td><input name=email type=text size=30 maxlength="200"></td>
  </tr>
  </table>
  <input type="submit" value="Submit" name="Submit" />
</form>
```

If an unregistered email address is provided, the API does not respond with an error. Instead, it emails that address saying that the user is not registered. This response is intended to prevent attempts to mine constituents.

Maintaining Logged-in Status on the Convio System

One issue with any single sign-on system is maintaining the logged-in state on the two systems. Systems typically expire a session after some period of inactivity. On Convio-powered systems, this time limit is 15 minutes. To keep the Convio-powered session alive, the partner web site only needs to include an image tag that references a servlet on the Convio powered site:

```
<img src=http://www.foo.org/site/PixelServer />
```

This will render a 1x1 clear GIF image. It is desirable to only include this once the user has been logged in to the Convio powered system.

Because the session cookie has been pushed to the user's browser, any links to the insecure domain (for example, www.foo.org) will automatically connect to the correct session and, as long as the partner site maintains the logged in status, will operate seamlessly for the user. Links directly to the secure URL (for example, https://secure2.convio.net/foo) should not be used from the partner site as it is possible that a session cookie has not been pushed yet. If the partner site links to a page that should be secure, such as a donation form over the insecure domain, the Convio code will automatically redirect and push the secure session cookie at that time.

loginTest

This method returns a cons_id if the user is already logged in or could be logged in because of a "remember me" cookie. The method supports HTTP GET requests so that it can be invoked via a redirect.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=loginTest &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]

Usage Notes

The loginTest method allows you to determine whether a user is already logged in to the Convio system when that user establishes a session with your partner system or attempts to access restricted content.

Note: The loginTest method should generally be invoked on an insecure channel (using "http" instead of the "https" protocol listed in the Syntax descriptions) because that is the most likely channel for a session that has already been established.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<loginResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cons_id>100245</cons_id>
  <timestamp>1359890307</timestamp>
  <signature>7a77328fd68c20a1f2c1864dc0865683</signature>
  <token xsi:nil="true"/>
  <login_test_url>https://secure2.convio.net/organization/site/CRConsAPI?
v=1.0&amp;api_key=site_api_key&amp;method=loginTest</login_test_url>
</loginResponse>
```

JSON response

```
{"loginResponse":{"login_test_url":"https://secure2.convio.net/organization/
site/CRConsAPI?
v=1.0&api_key=site_api_key&method=loginTest","timestamp":"1359890307","token":null,"cons_
```

Examples

Establishing a Session

When a user comes to the partner system for the first time, the system establishes an initial session. The partner system then wants to check with Convio to determine if the user can be logged in with a "remember me" cookie, or whether the user is already logged in to the Convio system.

The solution in this scenario is to redirect the user to:

```
    http://[org]/site/CRConsAPI?
method=loginTest&api_key=(key)&v=1.0&sign_redirects=true&success_redirect=(page)?
cons_id=${loginResponse/cons_id}&error_redirect=(page)?cons_id=0
```

Upon the user's return, verify the signature and, if `cons_id != 0`, log in the user. If the cons_id on the return is zero, the partner system should treat that user as an anonymous user.

Accessing Restricted Content

When a user who is not logged in makes a request for content available only to registered users, the partner system will want to check with Convio to see either if the user can be logged in with a "remember me" cookie, or if the user has already logged in to the Convio system.

The solution in this scenario is to redirect the user to `http://[org]/site/CRConsAPI?method=loginTest&api_key=(key)&v=1.0&sign_redirects=true&success_redirect=(page)?cons_id=${loginResponse/cons_id}&error_redirect=(login page)`

Upon the user's return, verify the signature and, if `cons_id != 0`, log the user in. If the user is not already logged in, the redirect loads a login page on the partner system. That login page will then follow the steps for the next scenario.

Accessing Restricted Content Without Redirection

This is for the same use case as the **Accessing Restricted Content** example above, used when the partner system needs to test asynchronously, without waiting on HTTP redirects.

The solution in this scenario is to invoke the API from the browser using JavaScript. The Convio Cross Domain Library can be used to facilitate this.

Upon the completion of the API call, if the `cons_id` field of the response is not zero, then use JavaScript to redirect the browser to the partner site. Pass the cons_id, timestamp, and signature fields from the response to the partner site's server so that the server can verify the authenticity of the request.

Verify the signature the same way you verify signed redirects. To verify the signature, append the cons_id, timestamp, and Convio API secret key into a single text string (without any delimiters between them) and compute its hash value. This value should match the signature provided in the API response. Remember that to prevent "replay" attacks, you should check that the timestamp is within a few seconds of the current time.

The partner site may want to continuously check whether the user has logged in to or out of the Convio-powered site without needing to redirect the browser when such an event occurs. The loginTest method response includes a URL that can be called from the partner site's server, enabling the server to customize the page for the user before sending it to the browser. It is important to note that the *initial* call **must** come from the browser to establish a session for the user and to create a URL that is specific to that session.

logout

This method logs out the currently-logged-in user.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=logout &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]

Usage Notes

The `logout` method closes a user's logged-in session with the Convio server.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<loginResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <message>User logged out successfully.</message>
</loginResponse>
```

JSON response

```
{"loginResponse":{"message":"User logged out successfully."}}
```

Examples

When a user wants to log out of the partner system, they expect to be logged out of the Convio system at the same time.

The solution in this scenario is to design a secure form that uses POST to this URL: `https://securex.convio.net/[org]/site/CRConsAPI`

The form should look something like this:

```
<form method="post" action="https://securex.convio.net/yoursite/site/
CRConsAPI">
  <input type="hidden" name="api_key" id="api_key" value="open" />
  <input type="hidden" name="v" id="v" value="1.0" />
  <input type="hidden" name="method" id="method" value="logout" />
  <input type="hidden" name="success_redirect" value="http://
yourothersite.org/logout.html" >
  <input type="hidden" name="error_redirect" value="http://yourothersite.org/
logout.html?code=${errorResponse/code}&message=${errorResponse/message}" >
  <input type="hidden" name="sign_redirects" id="sign_redirects"
 value="true" />
  <input type="submit" value="Logout" name="Submit" />
</form>
```

Upon the user's return, partner system should process its logout and then redirect the user to an appropriate page.

singleSignOn

This method uses a single sign-on token to log the site visitor in to the site as a registered user. The user identification is embedded in the token. The token is obtained by the external server by calling the getSingleSignOnToken, login, or authenticateUser API methods. Use this method when calling from a Web Client.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRConsAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRConsAPI?method=singleSignOn`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&sso_auth_token=`*value* [`&remember_me=`*value*]

Usage Notes

This **client-only** method logs in a user authenticated by an external system, by passing a valid sso_auth_token returned by *getSinglSignOnToken*.

Note: Use this API only if the third-party server will authenticate the user. To use the Convio server as the master authenticator, use *login* or *authenticateUser* instead.

See *Single Sign On (Convio as client)* for additional information on these API methods.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

remember_me

Optional. If set, a log-in cookie is pushed to the browser to recognize the current user on his next visit.

Type xsd:boolean.

Default is false.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<singleSignOnMethodResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">9dlp0oxn21.app8007</singleSignOnMethodResponse>
```

JSON response

```
{"singleSignOnMethodResponse":{"content":"9dlp0oxn21.app8007"}}
```

Survey API

CRSurveyAPI

`https://secure2.convio.net/organization/site/CRSurveyAPI`

Description

The Survey API provides a way for clients to retrieve a list of available surveys, the details and questions of a specific survey, and submit a constituent's response to a survey.

The *listSurveys* method is the only method in this API that does not require an *Authentication Token* for the current user session, either from an embedded [[S86]] session tag in a Convio web page, or returned by the appropriate *Single Sign-On API*. All other methods require this authentication token.

All of the methods in the Survey API automatically include permission checking such that only the surveys or questions that the user has permission to see will be displayed to them, just as they are when viewing the survey list or a survey embedded in a PageBuilder page.

The Survey API provides the following methods:

getSurvey

Returns a single active (published and not expired) survey that the user has permission to view, with the included (requested) questions that the user has permission to view.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRSurveyAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRSurveyAPI?method=getSurvey &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&survey_id=`*value* [`&auth=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

Use this **client-only** method to retrieve all the information you need in order to present an active survey for a user to fill out, generally when you intend to submit the user's response to that survey via the *submitSurvey* method.

Note: Only active Surveys can be retrieved with this method, meaning a survey that is published and has not expired.

Note: Only Surveys the user has permission to view will be retrieved by this method, so an unregistered user visiting a page that calls this method to retrieve a survey in the "Registered Users Only" category will not receive a survey in response to the call.

Note: Only questions the user has permission to view will be retrieved, so for a survey containing questions for all users and questions for "Registered Users Only", a user who is not a Registered User will not receive the "Registered Users Only" questions.

Questions that are configured to update a constituent record will contain the value that should be pre-filled for that question as part of the questionRecord. For Constituent Registration Information Questions, the value that should be pre-filled for a given field is contained as part of the questionTypeData.ConsRegData.contactInfoField for that specific field. Since this method can be used to retrieve constituent information, it can can only be called from trusted sources and requires that the caller pass an authentication token.

Implementation Tips

This method requires authentication, meaning that you must provide either an [[S86]] generated auth token in the auth parameter or a single sign on authentication token in the sso_auth_token parameter. For more information on authentication tokens, please refer to *Authentication Token*.

It is particularly important to understand the "Calling the API from an Insecure Page" section that Authentication Tokens page. Using the authentication token requires that the session in which the token is generated and the session in which the token is validated be the same session, otherwise the authentication check fails and you will receive a "Method Not Allowed" error.

This can happen when you obtain an authentication token for an insecure session (such as on an insecure PageBuilder page) and then make a call to getSurvey using the secure Survey API url. In order for this to work successfully, you must append the session ID to the request to the Secure URL in order to ensure that the user's authenticated session information is transferred from the insecure server to the secure server. This will allow the authentication token retrieved on the insecure server to be validated by the secure server.

If you are using PageBuilder, you can use the T2 tag to wrap the URL used to make the getSurvey call as described in the Authentication Tokens documentation and use the S86 tag to generate the authentication token.

If you cannot use the T2 tag (which is specific to Convio-hosted pages), you will still need to ensure that the user's authenticated session is transferred from the insecure server to the secure server. One of the easy ways to do this is to make a call to one of the Secure Servlets in Convio but use the insecure URL. For example: `http://demo829.convio.net/site/AjaxHelper` (replace the "demo829.convio.net" with your site's domain name). The AjaxHelper servlet is a secure servlet, so the Convio system will recognize that you have requested a secure servlet using the insecure URL and will redirect you to the appropriate secure servlet on the secure url, appending the session information necessary to transfer the session.

In order for this to work, you would need to first make the insecure request to the AjaxHelper servlet and allow that request and subsequent redirect to complete successfully, which establishes and transfers the session on the insercure server to the secure server. This gives you a session on both servers that can be recognized by the system. Once that is complete, you can obtain the Authentication Token, then make your call to getSurvey using that authentication token. For one example of how to do this, see the *Survey API Reference Implementation*.

Captcha Questions

These questions require some special handling in order to remain accessible and to avoid preventing submission of your surveys. Captcha works by requesting a challenge object, which is either an image or a sound file, and the user must enter the correct answer to that challenge object in order to "pass" the Captcha. Which challenge object was received is stored in the user's session, and the user must answer it on the next request or the object is expired and an answer to that object will no longer be valid. This means that when dealing with Captcha Questions you must request a new challenge object any time the next request does not contain the correct answer to the captcha question. This includes:

- The user refreshing the page containing your survey.
- The user submitting the survey and receiving any errors.

For accessibility, you must provide the user with an option to switch from an image based challenge object to an audio file challenge object. The switch must include requesting a new challenge object. The following example uses the *YUI Library* to accomplish this:

```
        /**
         * Registers a CAPTCHA question in global storage so that in the
         * event of any error, we can update the Captcha Image.
         *
         * Note that the way CAPTCHA works, only the last question
         * to be rendered will actually work.  Including multiple CAPTCHA
         * questions in a survey serves no purpose, but it is possible.
         *
         * @param the questionRecord from the getSurveyResponse document to
         *      be registered
         */
        var registerCaptchaQuestion = function(question){
            YAHOO.Convio.Survey.Data.captchaQuestion = question;
        }
        /**
         * Changes the Captcha Image for the provided question.
         * @param event JavaScript Event that triggered this call (generally a
Click event)
         * @param question the captcha question to change the image for
         */
        var changeCaptchaImage = function(event, question) {
            var captchaData = question.questionTypeData.captchaData;
```

```
            var ts = new Date().getTime();
            document.getElementById('captcha_img_' +
question.questionId).src=captchaData.imageSource+'?ts='+ts;
            YAHOO.util.Dom.removeClass(document.getElementById('captcha_img_'
+ question.questionId), "hidden-form");
            YAHOO.util.Dom.addClass(document.getElementById('captcha_player_'
+ question.questionId), "hidden-form");
        }

        /**
         * Resets the captcha question registered in global storage to remove
any value from it's
         * input field and changes the image to a new image.
         *
         * Call this upon an unsuccessful attempt to submit the survey in
order to reset the captcha
         * question for the user to answer again.
         */
        var resetCaptchaQuestion = function(){

if(YAHOO.Convio.Survey.Utils.hasValue(YAHOO.Convio.Survey.Data.captchaQuestion))
{
                var inputEl = YAHOO.util.Dom.get("question_" +
YAHOO.Convio.Survey.Data.captchaQuestion.questionId);
                inputEl.value = "";
                changeCaptchaImage(null, question);
            }
        }

        /**
         * Renders a Captcha Question to the DOM as a child of the provided
parentEl
         *
         * @param parentEl HTML element that this question should be appended
to
         * @param question questionRecord for a Captcha question from the
getSurveyResponse
         */
        var renderCaptchaQuestion = function(parentEl, question) {
            // register the question in Global storage:
            registerCaptchaQuestion(question);

            // Create a Div to hold this question and it's components:
            var qDiv = document.createElement("div");
            qDiv.id = "container_question_" + question.questionId;
            YAHOO.util.Dom.addClass(qDiv, "question-container");

            // Add the question's label (Question Text)
            var labelDiv = document.createElement("div");
            YAHOO.util.Dom.addClass(labelDiv, "question-text");
            if(question.questionRequired == 'true'){
                var reqElem = document.createElement("span");
                YAHOO.util.Dom.addClass(reqElem, "question-required");
                reqElem.appendChild(document.createTextNode('* '));
                labelDiv.appendChild(reqElem);
            }

            // EXAMPLE NOTES: YAHOO.Convio.Survey.Utils.ensureArray is a
utility function
            //    to ensure that the data passed in is an array, the details
of which
            //    are not included in this example.
            //
```

```
            //    questionNumber is a global variable that is incremented each
 time a
            //    question is numbered.
            if(YAHOO.Convio.Survey.Data.Survey.isNumberQuestions == 'true'
               &&
 YAHOO.Convio.Survey.Utils.ensureArray(YAHOO.Convio.Survey.Data.Survey.surveyQuestions).l
 > 1){
                var numberDiv = document.createElement("span");
                YAHOO.util.Dom.addClass(numberDiv, "question-number");
                numberDiv.appendChild(document.createTextNode(questionNumber++
 + ". "));
                elem.appendChild(numberDiv);
            }

            // add an HTML Label Element for the input field:
            var labelEl = document.createElement("label");
            YAHOO.util.Dom.addClass(labelEl, "question-label");
            labelEl.id = "label_" + question.questionId;
            YAHOO.util.Dom.setAttribute(labelEl, "for", "question_" +
 question.questionId);

 labelEl.appendChild(document.createTextNode(question.questionText));
            labelDiv.appendChild(labelEl);
            qDiv.appendChild(labelDiv);

            var inputDiv = document.createElement("div");
            YAHOO.util.Dom.addClass(inputDiv, "question-input");

            var captchaContainer = document.createElement("div");
            captchaContainer.id = "captcha-input-container";
            YAHOO.util.Dom.addClass(captchaContainer, "captcha-container");

            var captchaInputContainer = document.createElement("div");
            captchaInputContainer.id = "captcha-input";
            YAHOO.util.Dom.addClass(captchaInputContainer, "captcha-input-
container");

            var inputEl = document.createElement("input");
            inputEl.type = "text";
            inputEl.id = "question_" + question.questionId;
            inputEl.name = "question_" + question.questionId;
            YAHOO.util.Dom.setAttribute(inputEl, "size", "10");
            YAHOO.util.Dom.setAttribute(inputEl, "maxlength", "20");

            var captchaData = question.questionTypeData.captchaData;
            // Create a Link that the user can click on to get an audio file
 for the challenge object:
            var audioLink = document.createElement("a");
            YAHOO.util.Dom.addClass(audioLink, "captcha-question-audiolink");
            YAHOO.util.Dom.setAttribute(audioLink, "href",
 "javascript:void(0);");
            YAHOO.util.Dom.setAttribute(audioLink, "title",
 captchaData.audioLinkLabel);
            // EXAMPLE NOTE: the src of the img should be relative to point to
 the image
            //     at this location; this image at this location is included
 by default on Convio
            //     sites.
            audioLink.innerHTML = "<img src='http://yourdomain.org/images/
Action_buttons/accessibility.gif' alt='" + captchaData.audioLinkLabel + "'>";

            // Create a link for the user to change the image if they need to
            var changeImageLink = document.createElement("a");
```

```
            YAHOO.util.Dom.addClass(changeImageLink, "captcha-question-
changeImageLink");
            YAHOO.util.Dom.setAttribute(changeImageLink, "href",
 "javascript:void(0);");
            YAHOO.util.Dom.setAttribute(changeImageLink, "title",
 captchaData.changeImageLabel);
            changeImageLink.innerHTML = captchaData.changeImageLabel;

            var breakTag = document.createElement("br");
            captchaInputContainer.appendChild(inputEl);
            captchaInputContainer.appendChild(audioLink);
            captchaInputContainer.appendChild(breakTag);
            captchaInputContainer.appendChild(changeImageLink);

            var captchaImageContainer = document.createElement("div");
            captchaImageContainer.id = "captcha-image";
            YAHOO.util.Dom.addClass(captchaImageContainer, "captcha-image-
container");

            var captchaPlayer = document.createElement("div");
            captchaPlayer.id = "captcha_player_" + question.questionId;

            var captchaImage = document.createElement("img");
            captchaImage.id = "captcha_img_" + question.questionId;
            YAHOO.util.Dom.setAttribute(captchaImage, "src",
 captchaData.imageSource);
            YAHOO.util.Dom.setAttribute(captchaImage, "alt", "Captcha Image");

            captchaImageContainer.appendChild(captchaPlayer);
            captchaImageContainer.appendChild(captchaImage);

            captchaContainer.appendChild(captchaInputContainer);
            captchaContainer.appendChild(captchaImageContainer);
            inputDiv.appendChild(captchaContainer);
            qDiv.appendChild(inputDiv);
            parentEl.appendChild(qDiv);

            // Add Listeners to manage Captcha interactions:
            YAHOO.util.Event.addListener(changeImageLink, "click",
 changeCaptchaImage, question);

            /**
             * This function is responsible for retrieving a new audio file
 challenge object.
             * It presents the audio file as either a Windows Media Player or
 Quicktime object
             * based on the user's browser.
             */
            var audio_challenge = function() {
                var ts = new Date().getTime();
                YAHOO.util.Dom.addClass(document.getElementById('captcha_img_'
 + question.questionId), "hidden-form");

 YAHOO.util.Dom.removeClass(document.getElementById('captcha_player_' +
 question.questionId), "hidden-form");

                // If user agent is Safari or Opera, then use QuickTime,
 otherwise use Windows Media Player
                if(YAHOO.env.ua.webkit > 0 || YAHOO.env.ua.opera > 0) {
                    document.getElementById('captcha_player_' +
 question.questionId).innerHTML =  '<OBJECT '
                    + 'CLASSID="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B" '
                    + 'WIDTH="160" HEIGHT="50" '
```

```
                    + 'CODEBASE="http://www.apple.com/qtactivex/
qtplugin.cab">'
                    + '<param name="SRC" value="' + captchaData.audioLink +
 '&ts=' + ts + '" /> '
                    + '<PARAM name="AUTOPLAY" VALUE="true">'
                    + '<PARAM name="CONTROLLER" VALUE="false">'
                    + '<PARAM name="VOLUME" VALUE="100">'
                    + '<PARAM name="ENABLEJAVASCRIPT" VALUE="true">'
                    + '<PARAM name="TYPE" VALUE="audio/wav">'
                    + '<embed classid="clsid:02BF25D5-8C17-4B23-BC80-
D3488ABDDC6B"'
                    + 'name="sound"'
                    + 'id="sound"'
                    + 'src="' + captchaData.audioLink + '&ts=' + ts + '"'
                    + 'pluginspage="http://www.apple.com/quicktime/download/"'
                    + 'volume="100"'
                    + 'enablejavascript="true" '
                    + 'type="audio/wav" '
                    + 'height="50" '
                    + 'width="160"'
                    + 'autostart="true"'
                    + '> </embed>'
                    + '</OBJECT>';

                } else {
                    document.getElementById('captcha_player_' +
 question.questionId).innerHTML =
                    '<object height="50" width="160" classid="CLSID:22D6F312-
B0F6-11D0-94AB-0080C74C7E95" ' +
                    'codebase="https://activex.microsoft.com/activex/controls/
mplayer/en/nsmp2inf.cab#Version=5,1,52,701" ' +
                    'standby="Loading Microsoft# Windows# Media Player
 components..." ' +
                    'type="application/x-oleobject" align="middle"> ' +
                    '<param name="FileName" value="' + captchaData.audioLink +
 '&ts=' + ts + '" /> ' +
                    '<param name="AutoStart" value="True" /> ' +
                    '<param name="ShowStatusBar" value="True" /> ' +
                    '<param name="ShowPositionControls" value="False"> ' +
                    '<param name="Volume" value="1"> ' +
                    '<param name="Mute" value="False"> ' +
                    '<param name="DefaultFrame" value="mainFrame" />' +
                    '<embed id="captcha_embed_"' + question.questionId +
 ' type="application/x-mplayer2" pluginspage="http://www.microsoft.com/
Windows/MediaPlayer/" width="160" height="50"  showstatusbar="true" src="' +
 captchaData.audioLink + '&ts=' + ts + '"></embed>' +
                    '<br />' +
                    '<a href="' + captchaData.audioLink + '&ts=' + ts +
                    '" target="_blank" title="' + captchaData.newWindowLabel
 +'">' + captchaData.standAlonePlayerLabel + '</a>' +
                    '</object>';
                }
            }
            YAHOO.util.Event.addListener(audioLink, "click", audio_challenge);
        };
```

Note: The 'hidden-form' class that is added or removed in the example is a simple css class that defines "display: none;", so adding this class hides an element from view and removing this class displays the element.

Note: This example is not a cut-and-paste ready to use example, as it does not include all dependencies or global variables. Please see *Convio Open Downloads* to search for complete code examples.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

survey_id

Required. The ID of the survey you are accessing.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getSurveyResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <survey>
    <surveyId>1001</surveyId>
    <surveyName>New User Survey</surveyName>
    <viewSecurityCategory>
      <id>1</id>
      <label>General</label>
    </viewSecurityCategory>
    <reportSecurityCategory>
      <id>2</id>
      <label>Administrators Only</label>
    </reportSecurityCategory>
    <surveyIntroduction>Welcome to our site!  Please take a few moments to
 tell us about your experience:</surveyIntroduction>
    <resetButtonLabel>Clear Response</resetButtonLabel>
    <skipButtonLabel>No, Thanks</skipButtonLabel>
    <submitButtonLabel>Tell us</submitButtonLabel>
    <isAllowMultipleSubmission>false</isAllowMultipleSubmission>
    <isSecureSurvey>false</isSecureSurvey>
    <isNumberQuestions>true</isNumberQuestions>
    <submitSurveyUrl>http://www.yourdomain.org/site/PageServer?
pagename=thank_you</submitSurveyUrl>
    <cancelSurveyUrl>http://www.yourdomain.org/site/PageServer</
cancelSurveyUrl>
    <state>PUBLISHED</state>
    <publishedDate>2013-02-03T05:18:27.146-06:00</publishedDate>
    <stopDate>2018-02-03T05:18:27.146-06:00</stopDate>
    <surveyQuestions>
      <questionText>Tell us about yourself:</questionText>
      <questionOrderNumber>1</questionOrderNumber>
```

```
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1001</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>ConsQuestion</questionType>
      <questionRequired>true</questionRequired>
      <questionTypeData>
        <consRegInfoData>
          <contactInfoField>
            <fieldName>cons_title</fieldName>
            <label>Title</label>
            <fieldStatus>REQUESTED</fieldStatus>
            <fieldOptionValues>
              <label/>
              <value/>
            </fieldOptionValues>
            <fieldOptionValues>
              <label>Mr.</label>
              <value>Mr.</value>
            </fieldOptionValues>
            <fieldOptionValues>
              <label>Ms.</label>
              <value>Ms.</value>
            </fieldOptionValues>
            <fieldOptionValues>
              <label>Mrs.</label>
              <value>Mrs.</value>
            </fieldOptionValues>
            <fieldOptionValues>
              <label>Miss</label>
              <value>Miss</value>
            </fieldOptionValues>
            <fieldOptionValues>
              <label>Dr.</label>
              <value>Dr.</value>
            </fieldOptionValues>
          </contactInfoField>
          <contactInfoField>
            <fieldName>cons_first_name</fieldName>
            <label>First</label>
            <fieldStatus>REQUIRED</fieldStatus>
            <prefillValue>J.</prefillValue>
          </contactInfoField>
          <contactInfoField>
            <fieldName>cons_last_name</fieldName>
            <label>Last</label>
            <fieldStatus>REQUIRED</fieldStatus>
            <prefillValue>Constituent</prefillValue>
          </contactInfoField>
          <contactInfoField>
            <fieldName>cons_email</fieldName>
            <label>Email:</label>
            <fieldStatus>REQUIRED</fieldStatus>
          </contactInfoField>
          <email_auto_opt_in_text xsi:nil="true"/>
          <full_name_row_label>Name:</full_name_row_label>
          <city_state_zip_row_label/>
          <layout>STANDARD</layout>
          <loginPrompt>NONE</loginPrompt>
          <passwordComponent>
            <instructions>Please enter a user name and password for logging
in when you return.  You can use this password to update your information or
receive personalized content.</instructions>
```

```
            <userName>
              <fieldName>cons_user_name</fieldName>
              <label>Username:</label>
              <fieldStatus>REQUIRED</fieldStatus>
              <minLength>5</minLength>
              <maxLength>60</maxLength>
            </userName>
            <userNameHint>5 to 60 characters</userNameHint>
            <password>
              <fieldName>cons_password</fieldName>
              <label>Password:</label>
              <fieldStatus>REQUIRED</fieldStatus>
              <minLength>5</minLength>
              <maxLength>20</maxLength>
            </password>
            <passwordHint>5 to 20 characters</passwordHint>
            <verifyPassword>
              <fieldName>cons_rep_password</fieldName>
              <label>Repeat Password:</label>
              <fieldStatus>REQUIRED</fieldStatus>
              <minLength>5</minLength>
              <maxLength>20</maxLength>
            </verifyPassword>
          </passwordComponent>
          <rememberMe>
            <fieldName>s_rememberMe</fieldName>
            <label>Remember me.</label>
            <checked>true</checked>
          </rememberMe>
        </consRegInfoData>
      </questionTypeData>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>We appreciate you taking the time to tell us about your
experience with our site.</questionText>
      <questionOrderNumber>2</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1002</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>Caption</questionType>
      <questionRequired>false</questionRequired>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>Why did you visit our website today?</questionText>
      <questionOrderNumber>3</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1003</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>ComboChoice</questionType>
      <questionRequired>true</questionRequired>
      <questionTypeData>
        <surveyQuestionData>
          <availableAnswer>
            <label>Curious about organization</label>
            <value>Curious about organization</value>
          </availableAnswer>
          <availableAnswer>
            <label>Needed information</label>
            <value>Needed information</value>
          </availableAnswer>
```

```
          <availableAnswer>
            <label>Wanted to take action</label>
            <value>Wanted to take action</value>
          </availableAnswer>
          <availableAnswer>
            <label>Referred by friend/colleague</label>
            <value>Referred by friend/colleague</value>
          </availableAnswer>
          <availableAnswer>
            <label>Wanted to donate</label>
            <value>Wanted to donate</value>
          </availableAnswer>
          <availableAnswer>
            <label>Seeking to participate in some way</label>
            <value>Seeking to participate in some way</value>
          </availableAnswer>
        </surveyQuestionData>
      </questionTypeData>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>Have you interacted with this organization previously to
visiting this website today?</questionText>
      <questionOrderNumber>4</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1004</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>MultiMulti</questionType>
      <questionRequired>true</questionRequired>
      <questionTypeData>
        <surveyQuestionData>
          <availableAnswer>
            <label>Yes, I have participated in the programs or received
services from this organization</label>
            <value>Yes, I have participated in the programs or received
services from this organization</value>
          </availableAnswer>
          <availableAnswer>
            <label>Yes, I have previously donated to this organization</label>
            <value>Yes, I have previously donated to this organization</value>
          </availableAnswer>
          <availableAnswer>
            <label>Yes, I am a paying member of this organization</label>
            <value>Yes, I am a paying member of this organization</value>
          </availableAnswer>
          <availableAnswer>
            <label>Yes, I have previously volunteered or worked with this
organization</label>
            <value>Yes, I have previously volunteered or worked with this
organization</value>
          </availableAnswer>
          <availableAnswer>
            <label>Yes, I have interacted in some other way</label>
            <value>Yes, I have interacted in some other way</value>
          </availableAnswer>
          <availableAnswer>
            <label>No, this is my first interaction</label>
            <value>No, this is my first interaction</value>
          </availableAnswer>
        </surveyQuestionData>
      </questionTypeData>
    </surveyQuestions>
    <surveyQuestions>
```

```
      <questionText>How many times have you used our website in the last
 month?</questionText>
      <questionOrderNumber>5</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1005</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>NumericValue</questionType>
      <questionRequired>true</questionRequired>
      <prefillValue>5</prefillValue>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>Was it easy to find what you were looking for on the site?
</questionText>
      <questionOrderNumber>6</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1006</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>YesNo</questionType>
      <questionRequired>true</questionRequired>
      <questionTypeData>
        <surveyQuestionData>
          <availableAnswer>
            <label>Yes</label>
            <value>Yes</value>
          </availableAnswer>
          <availableAnswer>
            <label>No</label>
            <value>No</value>
          </availableAnswer>
        </surveyQuestionData>
      </questionTypeData>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>If you had difficulty finding information, what could we
 change to make it easier to find?</questionText>
      <questionOrderNumber>7</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1007</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>TextValue</questionType>
      <questionRequired>false</questionRequired>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>How would you rate the visual appeal of our site?</
questionText>
      <questionOrderNumber>8</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1008</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>RatingScale</questionType>
      <questionRequired>false</questionRequired>
      <questionTypeData>
        <surveyQuestionData>
          <availableAnswer>
            <label>Good</label>
            <value>2</value>
```

```
          </availableAnswer>
          <availableAnswer>
            <label>Excellent</label>
            <value>1</value>
          </availableAnswer>
          <availableAnswer>
            <label>Neither good nor bad</label>
            <value>3</value>
          </availableAnswer>
          <availableAnswer>
            <label>Bad</label>
            <value>4</value>
          </availableAnswer>
          <availableAnswer>
            <label>Worst</label>
            <value>5</value>
          </availableAnswer>
        </surveyQuestionData>
      </questionTypeData>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>Which section of our site did you find the most valuable?
</questionText>
      <questionOrderNumber>9</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1009</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>MultiSingleRadio</questionType>
      <questionRequired>false</questionRequired>
      <questionTypeData>
        <surveyQuestionData>
          <availableAnswer>
            <label>News</label>
            <value>News</value>
          </availableAnswer>
          <availableAnswer>
            <label>Programs</label>
            <value>Programs</value>
          </availableAnswer>
          <availableAnswer>
            <label>Events</label>
            <value>Events</value>
          </availableAnswer>
          <availableAnswer>
            <label>Community</label>
            <value>Community</value>
          </availableAnswer>
          <availableAnswer>
            <label>About Us</label>
            <value>About Us</value>
          </availableAnswer>
        </surveyQuestionData>
      </questionTypeData>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>If you could add a section to our site, what would it be?
</questionText>
      <questionOrderNumber>10</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1010</questionId>
      <categoryId>1</categoryId>
```

```
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>ShortTextValue</questionType>
      <questionRequired>false</questionRequired>
      <prefillValue>Daily Funnies</prefillValue>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>What information would you like to have included in that
 section?</questionText>
      <questionOrderNumber>11</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1011</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>LargeTextValue</questionType>
      <questionRequired>false</questionRequired>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>This survey is annoying you:</questionText>
      <questionOrderNumber>12</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1012</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>TrueFalse</questionType>
      <questionRequired>false</questionRequired>
      <questionTypeData>
        <surveyQuestionData>
          <availableAnswer>
            <label>True</label>
            <value>True</value>
          </availableAnswer>
          <availableAnswer>
            <label>False</label>
            <value>False</value>
          </availableAnswer>
        </surveyQuestionData>
      </questionTypeData>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>What types of email are you interested in receiving?</
questionText>
      <questionOrderNumber>13</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1013</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>Categories</questionType>
      <questionRequired>false</questionRequired>
      <questionTypeData>
        <surveyQuestionData>
          <availableAnswer>
            <label>News and Information</label>
            <value>1013</value>
            <selected>true</selected>
          </availableAnswer>
          <availableAnswer>
            <label>Events</label>
            <value>1041</value>
            <selected>true</selected>
          </availableAnswer>
          <availableAnswer>
```

```
            <label>Advocacy Alerts</label>
            <value>2012</value>
            <selected>true</selected>
          </availableAnswer>
        </surveyQuestionData>
      </questionTypeData>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>What would be the best day of the week to email you?</
questionText>
      <questionOrderNumber>14</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1014</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>MultiSingle</questionType>
      <questionRequired>true</questionRequired>
      <questionTypeData>
        <surveyQuestionData>
          <availableAnswer>
            <label>Sunday</label>
            <value>Sunday</value>
          </availableAnswer>
          <availableAnswer>
            <label>Monday</label>
            <value>Monday</value>
          </availableAnswer>
          <availableAnswer>
            <label>Tuesday</label>
            <value>Tuesday</value>
          </availableAnswer>
          <availableAnswer>
            <label>Wednesday</label>
            <value>Wednesday</value>
          </availableAnswer>
          <availableAnswer>
            <label>Thursday</label>
            <value>Thursday</value>
          </availableAnswer>
          <availableAnswer>
            <label>Friday</label>
            <value>Friday</value>
          </availableAnswer>
          <availableAnswer>
            <label>Saturday</label>
            <value>Saturday</value>
          </availableAnswer>
        </surveyQuestionData>
      </questionTypeData>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>Add Interests</questionText>
      <questionOrderNumber>15</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1015</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>HiddenInterests</questionType>
      <questionRequired>false</questionRequired>
      <questionTypeData>
        <surveyQuestionData>
          <availableAnswer>
```

```
            <label>Feedback</label>
            <value>1010</value>
          </availableAnswer>
        </surveyQuestionData>
      </questionTypeData>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>What day should we deliver your free tote bag for taking
this survey?</questionText>
      <questionOrderNumber>16</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1016</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>DateQuestion</questionType>
      <questionRequired>false</questionRequired>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>Blue</questionText>
      <questionOrderNumber>17</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1017</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>HiddenTextValue</questionType>
      <questionRequired>false</questionRequired>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>TRUE</questionText>
      <questionOrderNumber>18</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1018</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>HiddenTrueFalse</questionType>
      <questionRequired>false</questionRequired>
      <questionTypeData>
        <surveyQuestionData>
          <availableAnswer>
            <label>True</label>
            <value>True</value>
          </availableAnswer>
          <availableAnswer>
            <label>False</label>
            <value>False</value>
          </availableAnswer>
        </surveyQuestionData>
      </questionTypeData>
    </surveyQuestions>
    <surveyQuestions>
      <questionText>Verify that you're human by entering the letters from the
image below:</questionText>
      <questionOrderNumber>19</questionOrderNumber>
      <questionMinResponses>0</questionMinResponses>
      <questionMaxResponses>0</questionMaxResponses>
      <questionId>1019</questionId>
      <categoryId>1</categoryId>
      <surveyInstanceId>1001</surveyInstanceId>
      <questionType>Captcha</questionType>
      <questionRequired>true</questionRequired>
      <questionTypeData>
```

```
        <captchaData>
          <audioLink>http://www.yourdomain.org/site/Captcha?
type=audio&amp;JServSessionIdr005=adslkna787dahjy</audioLink>
          <imageSource>http://www.yourdomain.org/site/Captcha</imageSource>
          <audioLinkLabel>Visually impaired? Click here to have an audio
 challenge played.  You will then need to enter the code that is spelled
 out.</audioLinkLabel>
          <changeImageLabel>Change image</changeImageLabel>
          <newWindowLabel>Opens new window.</newWindowLabel>
          <standAlonePlayerLabel>Click here for standalone player</
standAlonePlayerLabel>
        </captchaData>
      </questionTypeData>
    </surveyQuestions>
  </survey>
</getSurveyResponse>
```

JSON response

```
{"getSurveyResponse":{"survey":
{"stopDate":"2018-02-03T05:18:27.152-06:00","resetButtonLabel":"Clear
 Response","surveyIntroduction":"Welcome to our site!
  Please take a few moments to tell us about your
 experience:","state":"PUBLISHED","surveyName":"New User
 Survey","submitSurveyUrl":"http://www.yourdomain.org/site/PageServer?
pagename=thank_you","skipButtonLabel":"No, Thanks","cancelSurveyUrl":"http://
www.yourdomain.org/site/PageServer","reportSecurityCategory":
{"id":"2","label":"Administrators Only"},"submitButtonLabel":"Tell
 us","isAllowMultipleSubmission":"false","surveyQuestions":
[{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"true","questi
 us about
 yourself:","questionOrderNumber":"1","questionType":"ConsQuestion","surveyInstanceId":"1
{"consRegInfoData":{"email_auto_opt_in_text":null,"rememberMe":
{"label":"Remember
 me.","fieldName":"s_rememberMe","checked":"true"},"layout":"STANDARD","loginPrompt":"NON
[{"fieldStatus":"REQUESTED","fieldOptionValues":[{"value":{},"label":
{}},{"value":"Mr.","label":"Mr."},{"value":"Ms.","label":"Ms."},
{"value":"Mrs.","label":"Mrs."},{"value":"Miss","label":"Miss"},
{"value":"Dr.","label":"Dr."}],"label":"Title","fieldName":"cons_title"},
{"fieldStatus":"REQUIRED","label":"First","fieldName":"cons_first_name","prefillValue":"J
{"fieldStatus":"REQUIRED","label":"Last","fieldName":"cons_last_name","prefillValue":"Con
{"fieldStatus":"REQUIRED","label":"Email:","fieldName":"cons_email"}],"passwordComponent"
{"userNameHint":"5 to 60 characters","passwordHint":"5 to 20
 characters","instructions":"Please enter a user name and password
 for logging in when you return.  You can use this password to update
 your information or receive personalized content.","userName":
{"fieldStatus":"REQUIRED","maxLength":"60","minLength":"5","label":"Username:","fieldName
{"fieldStatus":"REQUIRED","maxLength":"20","minLength":"5","label":"Password:","fieldName
{"fieldStatus":"REQUIRED","maxLength":"20","minLength":"5","label":"Repeat
 Password:","fieldName":"cons_rep_password"}},"full_name_row_label":"Name:","city_state_z
{}}}},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
 appreciate you taking the time to tell us about your experience with our
 site.","questionOrderNumber":"2","questionType":"Caption","surveyInstanceId":"1001"},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"true","questio
 did you visit our website
 today?","questionOrderNumber":"3","questionType":"ComboChoice","surveyInstanceId":"1001"
{"surveyQuestionData":{"availableAnswer":[{"value":"Curious about
 organization","label":"Curious about organization"},{"value":"Needed
 information","label":"Needed information"},{"value":"Wanted to take
 action","label":"Wanted to take action"},{"value":"Referred by friend/
```

```
colleague","label":"Referred by friend/colleague"},{"value":"Wanted to
 donate","label":"Wanted to donate"},{"value":"Seeking to participate
 in some way","label":"Seeking to participate in some way"}]}}},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"true","questio
 you interacted with this organization previously to visiting this website
 today?","questionOrderNumber":"4","questionType":"MultiMulti","surveyInstanceId":"1001",
{"surveyQuestionData":{"availableAnswer":[{"value":"Yes, I have participated
 in the programs or received services from this organization","label":"Yes,
 I have participated in the programs or received services from this
 organization"},{"value":"Yes, I have previously donated to this
 organization","label":"Yes, I have previously donated to this organization"},
{"value":"Yes, I am a paying member of this organization","label":"Yes,
 I am a paying member of this organization"},{"value":"Yes, I have
 previously volunteered or worked with this organization","label":"Yes,
 I have previously volunteered or worked with this organization"},
{"value":"Yes, I have interacted in some other way","label":"Yes,
 I have interacted in some other way"},{"value":"No, this is my
 first interaction","label":"No, this is my first interaction"}]}}},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"true","questio
 many times have you used our website in the last
 month?","questionOrderNumber":"5","questionType":"NumericValue","surveyInstanceId":"1001
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"true","questio
 it easy to find what you were looking for on the
 site?","questionOrderNumber":"6","questionType":"YesNo","surveyInstanceId":"1001","quest
{"surveyQuestionData":{"availableAnswer":[{"value":"Yes","label":"Yes"},
{"value":"No","label":"No"}]}}},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
 you had difficulty finding information, what could we change to make it
 easier to
 find?","questionOrderNumber":"7","questionType":"TextValue","surveyInstanceId":"1001"},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
 would you rate the visual appeal of our
 site?","questionOrderNumber":"8","questionType":"RatingScale","surveyInstanceId":"1001",
{"surveyQuestionData":{"availableAnswer":[{"value":"2","label":"Good"},
{"value":"1","label":"Excellent"},{"value":"3","label":"Neither good nor
 bad"},{"value":"4","label":"Bad"},{"value":"5","label":"Worst"}]}}},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
 section of our site did you find the most
 valuable?","questionOrderNumber":"9","questionType":"MultiSingleRadio","surveyInstanceI
{"surveyQuestionData":{"availableAnswer":[{"value":"News","label":"News"},
{"value":"Programs","label":"Programs"},{"value":"Events","label":"Events"},
{"value":"Community","label":"Community"},{"value":"About Us","label":"About
 Us"}]}}},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
 you could add a section to our site, what would it
 be?","questionOrderNumber":"10","questionType":"ShortTextValue","surveyInstanceId":"1001
 Funnies"},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
 information would you like to have included in that
 section?","questionOrderNumber":"11","questionType":"LargeTextValue","surveyInstanceId"
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
 survey is annoying
 you:","questionOrderNumber":"12","questionType":"TrueFalse","surveyInstanceId":"1001","q
{"surveyQuestionData":{"availableAnswer":[{"value":"True","label":"True"},
{"value":"False","label":"False"}]}}},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
 types of email are you interested in
 receiving?","questionOrderNumber":"13","questionType":"Categories","surveyInstanceId":"1
{"surveyQuestionData":{"availableAnswer":
[{"selected":"true","value":"1013","label":"News and
 Information"},{"selected":"true","value":"1041","label":"Events"},
{"selected":"true","value":"2012","label":"Advocacy Alerts"}]}}},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"true","questio
 would be the best day of the week to email
```

```
 you?","questionOrderNumber":"14","questionType":"MultiSingle","surveyInstanceId":"1001",
{"surveyQuestionData":{"availableAnswer":[{"value":"Sunday","label":"Sunday"},
{"value":"Monday","label":"Monday"},{"value":"Tuesday","label":"Tuesday"},
{"value":"Wednesday","label":"Wednesday"},
{"value":"Thursday","label":"Thursday"},{"value":"Friday","label":"Friday"},
{"value":"Saturday","label":"Saturday"}]}}},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
 Interests","questionOrderNumber":"15","questionType":"HiddenInterests","surveyInstanceId
{"surveyQuestionData":{"availableAnswer":
{"value":"1010","label":"Feedback"}}}},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
 day should we deliver your free tote bag for taking this
 survey?","questionOrderNumber":"16","questionType":"DateQuestion","surveyInstanceId":"10
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"false","questi
{"surveyQuestionData":{"availableAnswer":[{"value":"True","label":"True"},
{"value":"False","label":"False"}]}}},
{"questionMaxResponses":"0","questionMinResponses":"0","questionRequired":"true","questio
 that you're human by entering the letters from the image
 below:","questionOrderNumber":"19","questionType":"Captcha","surveyInstanceId":"1001","c
{"captchaData":{"newWindowLabel":"Opens new
 window.","changeImageLabel":"Change image","audioLinkLabel":"Visually
 impaired? Click here to have an audio challenge played.
  You will then need to enter the code that is spelled
 out.","imageSource":"http://www.yourdomain.org/site/
Captcha","audioLink":"http://www.yourdomain.org/site/Captcha?
type=audio&JServSessionIdr005=adslkna787dahjy","standAlonePlayerLabel":"Click
 here for standalone
 player"}}}],"isNumberQuestions":"true","publishedDate":"2013-02-03T05:18:27.152-06:00","
{"id":"1","label":"General"},"surveyId":"1001","isSecureSurvey":"false"}}}
```


Examples

To retrieve the survey with the survey_id=1021:

```
https://demo-secure.convio.net/demo829/site/CRSurveyAPI?
method=getSurvey&v=1.0&api_key=MyAPIKey&survey_id=1021
```

To embed a form on an insecure PageBuilder page that allows the user to request a specific survey:

```
 <form action="[[T2:https://[[S29:SECURE_DOMAIN]]
[[S29:SECURE_PATH]]CRSurveyAPI]]" method="post">
  <input name="api_key" type="hidden" value="MyAPIKey" />
  <input name="v" type="hidden" value="1.0" />
  <input name="method" type="hidden" value="getSurvey" />
  [[S86]]
  <label for="survey_id">Survey ID: </label>
  <input name="survey_id" type="text" value="1001" size="15" maxlength="6" />
  <input name="Submit" type="submit" value="Submit" />
 </form>
```

listSurveys

Returns list of Surveys the user has permission to view. List provides only the metadata about the survey, and does not include the questions for each survey.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRSurveyAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRSurveyAPI?method=listSurveys`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&published_only=`*value* [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*]

Usage Notes

Use this **client-only** method to retrieve a list of all the surveys available to that user. The user must have permission to view a survey in order for it to be returned by this method.

Note: Unpublished surveys are not accessible outside of the Convio Admin Interface, so you will almost always want to retrieve only the published surveys. For this reason the published_only parameter is optional and defaults to true.

Sorting or Filtering by Column:

To sort or filter the results, the list_sort_column and list_filter_column parameters accept the following values (case-sensitive):

- surveyId
- surveyName
- surveyIntroduction
- isAllowMultipleSubmission
- isSecureSurvey
- isNumberQuestions
- state
- publishedDate
- stopDate

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

published_only

Required. Parameter indicating whether or not to list only published surveys. Default is true.

Type xsd:boolean.

Default is true.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listSurveysResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <pagingMetadata>
    <currentSize>3</currentSize>
    <isFinalSize>true</isFinalSize>
  </pagingMetadata>
  <surveys>
    <surveyId>1001</surveyId>
    <surveyName>New User Survey</surveyName>
    <viewSecurityCategory>
      <id>1</id>
      <label>General</label>
    </viewSecurityCategory>
    <reportSecurityCategory>
      <id>2</id>
      <label>Administrators Only</label>
    </reportSecurityCategory>
    <surveyIntroduction>Welcome to our site!  Please take a few moments to
 tell us about your experience:</surveyIntroduction>
    <resetButtonLabel>Clear Response</resetButtonLabel>
    <skipButtonLabel>No, Thanks</skipButtonLabel>
    <submitButtonLabel>Tell us</submitButtonLabel>
    <isAllowMultipleSubmission>false</isAllowMultipleSubmission>
    <isSecureSurvey>false</isSecureSurvey>
    <isNumberQuestions>true</isNumberQuestions>
    <submitSurveyUrl>http://www.yourdomain.org/site/PageServer?
pagename=thank_you</submitSurveyUrl>
```

```
    <cancelSurveyUrl>http://www.yourdomain.org/site/PageServer</
cancelSurveyUrl>
    <state>PUBLISHED</state>
    <publishedDate>2013-02-03T05:18:27.064-06:00</publishedDate>
    <stopDate>2018-02-03T05:18:27.064-06:00</stopDate>
  </surveys>
  <surveys>
    <surveyId>1002</surveyId>
    <surveyName>January 2009 User Engagement</surveyName>
    <viewSecurityCategory>
      <id>1</id>
      <label>General</label>
    </viewSecurityCategory>
    <reportSecurityCategory>
      <id>2</id>
      <label>Administrators Only</label>
    </reportSecurityCategory>
    <surveyIntroduction/>
    <resetButtonLabel>Clear Response</resetButtonLabel>
    <skipButtonLabel>No, Thanks</skipButtonLabel>
    <submitButtonLabel>Tell us</submitButtonLabel>
    <isAllowMultipleSubmission>false</isAllowMultipleSubmission>
    <isSecureSurvey>false</isSecureSurvey>
    <isNumberQuestions>true</isNumberQuestions>
    <submitSurveyUrl>http://www.yourdomain.org/site/PageServer?
pagename=thank_you</submitSurveyUrl>
    <cancelSurveyUrl>http://www.yourdomain.org/site/PageServer</
cancelSurveyUrl>
    <state>PUBLISHED</state>
    <publishedDate>2013-02-03T05:18:27.064-06:00</publishedDate>
    <stopDate>2018-02-03T05:18:27.064-06:00</stopDate>
  </surveys>
  <surveys>
    <surveyId>1003</surveyId>
    <surveyName>Fun Event Survey</surveyName>
    <viewSecurityCategory>
      <id>1</id>
      <label>General</label>
    </viewSecurityCategory>
    <reportSecurityCategory>
      <id>2</id>
      <label>Administrators Only</label>
    </reportSecurityCategory>
    <surveyIntroduction/>
    <resetButtonLabel>Clear Response</resetButtonLabel>
    <skipButtonLabel>No, Thanks</skipButtonLabel>
    <submitButtonLabel>Tell us</submitButtonLabel>
    <isAllowMultipleSubmission>false</isAllowMultipleSubmission>
    <isSecureSurvey>false</isSecureSurvey>
    <isNumberQuestions>true</isNumberQuestions>
    <submitSurveyUrl>http://www.yourdomain.org/site/PageServer?
pagename=thank_you</submitSurveyUrl>
    <cancelSurveyUrl>http://www.yourdomain.org/site/PageServer</
cancelSurveyUrl>
    <state>PUBLISHED</state>
    <publishedDate>2013-02-03T05:18:27.065-06:00</publishedDate>
    <stopDate>2018-02-03T05:18:27.065-06:00</stopDate>
  </surveys>
</listSurveysResponse>
```

JSON response


```
{"listSurveysResponse":{"pagingMetadata":
{"isFinalSize":"true","currentSize":"3"},"surveys":
[{"stopDate":"2018-02-03T05:18:27.066-06:00","resetButtonLabel":"Clear
 Response","surveyIntroduction":"Welcome to our site!
  Please take a few moments to tell us about your
 experience:","state":"PUBLISHED","surveyName":"New User
 Survey","submitSurveyUrl":"http://www.yourdomain.org/site/PageServer?
pagename=thank_you","skipButtonLabel":"No, Thanks","cancelSurveyUrl":"http://
www.yourdomain.org/site/PageServer","reportSecurityCategory":
{"id":"2","label":"Administrators Only"},"submitButtonLabel":"Tell
 us","isAllowMultipleSubmission":"false","isNumberQuestions":"true","publishedDate":"2013
{"id":"1","label":"General"},"surveyId":"1001","isSecureSurvey":"false"},
{"stopDate":"2018-02-03T05:18:27.067-06:00","resetButtonLabel":"Clear
 Response","surveyIntroduction":{},"state":"PUBLISHED","surveyName":"January
 2009 User Engagement","submitSurveyUrl":"http://www.yourdomain.org/
site/PageServer?pagename=thank_you","skipButtonLabel":"No,
 Thanks","cancelSurveyUrl":"http://www.yourdomain.org/site/
PageServer","reportSecurityCategory":{"id":"2","label":"Administrators
 Only"},"submitButtonLabel":"Tell
 us","isAllowMultipleSubmission":"false","isNumberQuestions":"true","publishedDate":"2013
{"id":"1","label":"General"},"surveyId":"1002","isSecureSurvey":"false"},
{"stopDate":"2018-02-03T05:18:27.067-06:00","resetButtonLabel":"Clear
 Response","surveyIntroduction":{},"state":"PUBLISHED","surveyName":"Fun
 Event Survey","submitSurveyUrl":"http://www.yourdomain.org/site/PageServer?
pagename=thank_you","skipButtonLabel":"No, Thanks","cancelSurveyUrl":"http://
www.yourdomain.org/site/PageServer","reportSecurityCategory":
{"id":"2","label":"Administrators Only"},"submitButtonLabel":"Tell
 us","isAllowMultipleSubmission":"false","isNumberQuestions":"true","publishedDate":"2013
{"id":"1","label":"General"},"surveyId":"1003","isSecureSurvey":"false"}]}}
```


Examples

To retrieve a list of only published surveys:

```
     https://demo-secure.convio.net/demo829/site/CRSurveyAPI?
method=listSurveys&v=1.0&api_key=MyAPIKey&published_only=true
```

To retrieve a list of published and unpublished surveys, perhaps if you are integrating with a 3rd party CMS and want to provide a means of embedding a survey using the *getSurvey* method, but you want the CMS administrator to see all surveys, including ones that users will not be able to take:

```
     https://demo-secure.convio.net/demo829/site/CRSurveyAPI?
method=listSurveys&v=1.0&api_key=MyAPIKey&published_only=false
```

submitSurvey

Accepts responses to a single, active (published and not expired) survey and stores the responses, including creating or updating the constituent record if appropriate.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRSurveyAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRSurveyAPI?method=submitSurvey`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [

`&redirect=`*value* [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&survey_id=`*value* [`&auth=`*value*] [`&cons_birth_date=`*value*] [`&cons_city=`*value*] [`&cons_country=`*value*] [`&cons_county=`*value*] [`&cons_email=`*value*] [`&cons_email_format=`*value*] [`&cons_email_opt_in=`*value*] [`&cons_employer=`*value*] [`&cons_first_name=`*value*] [`&cons_gender=`*value*] [`&cons_last_name=`*value*] [`&cons_middle_name=`*value*] [`&cons_occupation=`*value*] [`&cons_password=`*value*] [`&cons_phone=`*value*] [`&cons_postal_opt_in=`*value*] [`&cons_prof_suffix=`*value*] [`&cons_rep_password=`*value*] [`&cons_state=`*value*] [`&cons_street1=`*value*] [`&cons_street2=`*value*] [`&cons_street3=`*value*] [`&cons_suffix=`*value*] [`&cons_title=`*value*] [`&cons_user_name=`*value*] [`&cons_zip_code=`*value*] [`&denyApiSubmit=`*value*] [`&question_=`*value*] [`&s_rememberMe=`*value*] [`&survey_cancel=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

Use this **client-only** method to submit a user's response to a single survey. To retrieve the survey, see the *getSurvey* method. Since this method can create and update constituent records it can only be used from trusted sources and requires an auth_token be passed in as the auth parameter.

Note: Only Active Surveys can be submitted to with this method, meaning a survey that is published and has not expired.

Note: Only Surveys the user has permission to view can be submitted to by this method.

Note: Only questions the user has permission to view will have their responses stored, even if you submit answers for other questions.

Note: See the "Implementation Tips" section of the *getSurvey* method for tips on handling Authentication Tokens, particularly if you get a "Method Not Allowed" error.

Submitting Responses To Questions

Submitting a response to a question is done by building a parameter using the question's ID and the prefix question_. For Example, to submit an answer of "test" to a question with the ID of 1002, you would use: "question_1002=test".

Note: The Hidden question types (Text, True/False, Interest Update) must still be submitted as part of this in order for the system to process the updates those questions represent.

Note: The question_ prefix is not used to submit responses to a Constituent Registration Information question. See the section below for handling those questions.

Constituent Registration Information (CRI) Questions

These questions are made up of multiple fields, and each field must be stored separately within Convio. For this reason, each field within a CRI question has it's own parameter that you use to submit the response for that field.

The processing of responses to CRI questions through the survey is the same as the processing through the normal Convio Survey application. The question's configuration within Convio determines whether to automatically log users in, how to handle cases where the user's responses match an existing constituent, etc.

Post-Submission Handling

Submitting a survey can login the user submitting the survey, you should refresh your auth token after submission.

This method will not automatically redirect to the URL specified in the Survey's configuration as the "Survey Submitted Page". The response will contain that URL as the nextUrl parameter if you do want to redirect the user there.

Note: If you submit a form directly to this method, the user cancelling the form will return a response where the "Survey Cancelled Page" is the nextUrl and the userCancelled is true.

The response also contains the Survey Thank You Page content that taking a survey directly (not through the API) would normally display. This is not a page that is possible to configure through the Convio Admin Interface, nor is it something that can be linked to directly, but if you want to provide the same user-experience through the API as what the user would see when taking a survey not through the API, this is the thank you page content they would see if there is no "Survey Submitted Page" set for the survey.

A successful response to this method will also contain the user's responses to questions that they can provide a meaningful answer to. The response will contain an array of userResponses that contain the questionID, Question Text, and the responseValues that a user provided to that question. This will let you build a custom thank you page that can echo back to the user the answers that they provided, or create branching functionality based on their answers in the event of a successful submission.

Note: The User Responses only contains answers to which the user can provide "meaningful" answers. The questions that are not echoed back are:

- Caption
- Hidden Interest Update
- Hidden Text Update
- Hidden True/False Update
- Captcha

Aside from Captcha questions, the user doesn't provide answers to the other questions, and the answer to a Captcha question is only meaningful in passing the Captcha challenge.

Note: The user responses will not contain the answers provided to a Constituent Registration Information question. If you would like to retrieve that data, you can use the *getUser* method to obtain the user's information.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

survey_id

Required. The ID of the survey you are accessing.

Type xsd:nonNegativeInteger.

cons_birth_date

Optional. Parameter containing the value of the Birth Date field when submitting a survey containing a Constituent Registration Information Question, formatted per the ISO 8601 specification.

Type xsd:dateTime.

cons_city

Optional. Parameter containing the value of the City field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_country

Optional. Parameter containing the value of the Country field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_county

Optional. Parameter containing the value of the County field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_email

Optional. Parameter containing the value of the Email field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_email_format

Optional. Parameter containing the value of the Preferred Email Format field when submitting a survey containing a Constituent Registration Information Question. Default value if not provided is '0' indicating the user has no preferred format.

Type xsd:nonNegativeInteger.

Default is 0.

cons_email_opt_in

Optional. Parameter indicating the Email Opt In choice of a user when submitting a survey containing a Constituent Registration Information (CRI) Question. True indicates the constituent should be opted in, false indicates they should be opted out.

Type xsd:boolean.

cons_employer

Optional. Parameter containing the value of the Employer field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_first_name

Optional. Parameter containing the value of the First Name field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_gender

Optional. Parameter containing the value of the Gender field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_last_name

Optional. Parameter containing the value of the Last Name field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_middle_name

Optional. Parameter containing the value of the Middle Name field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_occupation

Optional. Parameter containing the value of the Occupation field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_password

Optional. Parameter containing the value of the Password field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_phone

Optional. Parameter containing the value of the Phone Number field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_postal_opt_in

Optional. Parameter indicating the Postal Mail Opt In choice of a user when submitting a survey containing a Constituent Registration Information (CRI) Question. True indicates the constituent should be opted in, false indicates they should be opted out.

Type xsd:boolean.

cons_prof_suffix

Optional. Parameter containing the value of the Professional Suffix field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_rep_password

Optional. Parameter containing the value of the Confirm / Re-enter Password field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_state

Optional. Parameter containing the value of the State field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_street1

Optional. Parameter containing the value of the Street 1 field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_street2

Optional. Parameter containing the value of the Street 2 field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_street3

Optional. Parameter containing the value of the Street 3 field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_suffix

Optional. Parameter containing the value of the Suffix field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_title

Optional. Parameter containing the value of the Title field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_user_name

Optional. Parameter containing the value of the User name field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

cons_zip_code

Optional. Parameter containing the value of the Zip / Postal Code field when submitting a survey containing a Constituent Registration Information Question.

Type xsd:string.

denyApiSubmit

Optional. Parameter indicating that the submission should NOT be accepted. This is a weak method of preventing scripts from submitting spam via your surveys by including a text input field that is hidden using Cascading Style Sheets (CSS). Do not use a hidden input field for this, as most spam-scripts do not auto-populate hidden inputs. If a script automatically fills in that field and submits the survey, the submission will not be processed. It is recommended that you provide a label indicating that humans should not fill in this field so that humans using a screen reader know to leave that field empty.

Type xsd:string.

question_

Optional. This parameter is not used by itself, but is used in conjunction with the Question's Id to create a Question Indicator Parameter. You must append a question's question Instance ID after the underscore, at which point this parameter indicates the answer to that question. Ex: 'question_1001=hello' would indicate that the question with the id 1001 is being submitted with an answer of 'hello'. When providing answers to a DateQuestion question, this parameter must be a date string formatted per the ISO 8601 specification. When providing answers to a NumericValue question this parameter must be a numeric value. For all other question types, this parameter is a string. Note: This type of Question Indicator Parameter is not used for Constituent Registration Information questions.

Type xsd:string.

s_rememberMe

Optional. Parameter indicating that the system should remember the user after logging them in (if the question is configured to log new users in).

Type xsd:boolean.

survey_cancel

Optional. Parameter indicating that user clicked the cancel/skip button. Submitting a value of 'true' indicates that the user has cancelled and the API will return a response indicating that the submission was successful and that the user cancelled. The response will include URL to redirect to based on the survey configuration (if any). Does not perform an automatic redirect.

Type xsd:boolean.

Default is false.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<submitSurveyResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
  <userCanceled>false</userCanceled>
  <nextUrl xsi:nil="true"/>
  <thankYouPageContent><![CDATA[<div class="appArea"><p>Thank you for taking
 the Tell us A Little About You survey. Your input is very valuable to
 us in designing our program.</p><p>This organization often shares survey
 results with registered users. There are many benefits to being a registered
 user, and you can help us serve you more effectively by registering.</
p><p><a href="http://yourdomain.org/site/ConsProfileUser">Click here to
 register and receive personalized content and email updates.</a></p></
div><div class="appArea"><form action="http://yourdomain.org/site/PageServer"
 method="post"><p><input type="submit" name="go" id="go" value="Home"
 class="Button" /></p><input type="hidden" name="SURVEY_ID" id="SURVEY_ID"
 value="1001" /></form></div>]]></thankYouPageContent>
  <userResponses>
    <questionId>1001</questionId>
    <questionText>What is your favorite color?</questionText>
    <responseValue>Blue</responseValue>
  </userResponses>
  <userResponses>
    <questionId>1010</questionId>
    <questionText>When did you first visit our site?</questionText>
    <responseValue>Jan 10 2010</responseValue>
  </userResponses>
  <userResponses>
    <questionId>1010</questionId>
    <questionText>What types of content are you interested in?</questionText>
    <responseValue>1003</responseValue>
    <responseValue>1028</responseValue>
  </userResponses>
</submitSurveyResponse>
```

JSON response


```
{"submitSurveyResponse":{"thankYouPageContent":"<div class=\"appArea
\"><p>Thank you for taking the Tell us A Little About You survey. Your input
 is very valuable to us in designing our program.<\/p><p>This organization
 often shares survey results with registered users. There are many benefits
 to being a registered user, and you can help us serve you more effectively
 by registering.<\/p><p><a href=\"http://yourdomain.org/site/ConsProfileUser
\">Click here to register and receive personalized content and email
 updates.<\/a><\/p><\/div><div class=\"appArea\"><form action=\"http://
yourdomain.org/site/PageServer\" method=\"post\"><p><input type=\"submit
\" name=\"go\" id=\"go\" value=\"Home\" class=\"Button\" /><\/p><input
 type=\"hidden\" name=\"SURVEY_ID\" id=\"SURVEY_ID\" value=\"1001\" /><
\/form><\/div>","nextUrl":null,"userCanceled":"false","userResponses":
[{"questionId":"1001","questionText":"What is your favorite
 color?","responseValue":"Blue"},{"questionId":"1010","questionText":"When
 did you first visit our site?","responseValue":"Jan 10 2010"},
{"questionId":"1010","questionText":"What types of content are you interested
 in?","responseValue":["1003","1028"]}],"success":"true"}}
```


Examples

The example is for a survey with an ID of 1024 and contains the following questions:

Question ID	Question Type	Question Text	User's Response
1000	Constituent Registration Information	Please provide your name and email:	Kelly, Constituent, devnull +kconstituent@convio.com
1001	Short Text	What is your favorite Color?	Blue
1010	Numeric Value	How many hours per week do you spend online (personal and business)?	40
1024	Hidden True/False Update	*Example Note: question is hidden, so there is no text*	TRUE
1030	Multiple Choice, Multiple Response	What days do you use the internet the most?	Wednesday, Friday

The following string should be submitted using a POST method:

```
     https://demo-secure.convio.net/demo829/site/CRSurveyAPI?
method=submitSurvey&v=1.0&api_key=MyAPIKey&survey_id=1024&question_1001=Blue&question_101
+kconstituent@convio.com
```

TeamRaiser API

CRTeamraiserAPI
`https://secure2.convio.net/organization/site/CRTeamraiserAPI`

Description

The TeamRaiser API provides a way for clients to create custom experiences for their special events participants using participant-specific data.

Many of these APIs operate on behalf of a specific user, and require an *Authentication Token*, representing a logged-in user session for the TeamRaiser participant, either from an embedded [[S86]] session tag in a Convio web page, or returned by the appropriate *Single Sign-On API*.

Some TeamRaiser APIs impose additional access restrictions: some of the Team-related functions are available only to Team Captains; some of the Company-related functions are available only to the Company Coordinator. See the Usage Notes section for each API regarding special access restrictions.

Note: There are certain steps that must be taken in Convio to enable the use of the TeamRaiser APIs. First, the OCI site option must be set to TRUE in the TeamRaiser site options and the URL-specified levels checkbox must be enabled in the General Configuration Gift Options for the Online Giving application.

Note: TeamRaiser Server API methods are accessible only to white-listed servers. The API Administrator account used to call Server API methods must belong to an Administrative Security Group having "TeamRaiser API User" permission under the "TeamRaiser Management" permission type. The built-in "API Administrators" group does not have this permission by default.

The TeamRaiser API provides the following methods:

acknowledgeGifts

Clears the gift acknowledgment follow-up flag for the specified contact.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=acknowledgeGifts &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&contact_id=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

contact_id

Optional. Donor address book ID

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<acknowledgeGiftsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
  <contactId>182333</contactId>
</acknowledgeGiftsResponse>
```

JSON response

```
{"acknowledgeGiftsResponse":{"contactId":"182333","success":"true"}}
```

addDraft

Adds a draft

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=addDraft`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* `&layout_id=`*value* `&message_id=`*value* `&message_name=`*value* `&subject=`*value*
[`&message_body=`*value*] [`&recipients=`*value*] [`&save_template=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

layout_id

Required. The layout ID.

Type xsd:nonNegativeInteger.

message_id

Required. Message ID

Type xsd:nonNegativeInteger.

message_name

Required. The name of the message.

Type xsd:string.

subject

Required. The message subject.

Type xsd:string.

message_body

Optional. The message body.

Type xsd:string.

recipients

Optional. The message recipients list.

Type xsd:string.

save_template

Optional. Indicates whether or not a message should be considered a permanent template.

Type xsd:boolean.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<addDraftResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <message>
    <subject>example subject</subject>
    <messageBody>example message body</messageBody>
    <messageName>example message name</messageName>
    <layoutId>1234</layoutId>
  </message>
</addDraftResponse>
```

JSON response

```
{"addDraftResponse":{"message":{"messageBody":"example message
 body","subject":"example subject","layoutId":"1234","messageName":"example
 message name"}}}
```

addGift

Provides a simple method for TeamRaiser participants to record gifts and pledges.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	SRTeamraiserAPI	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=addGift &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&first_name=`*value* `&gift_amount=`*value* `&last_name=`*value* `&payment_type=`*cash* | *check* | *later* | *credit* | *ach* `&team_id=`*value* [`&billing_city=`*value*] [`&billing_county=`*value*] [`&billing_first_name=`*value*] [`&billing_last_name=`*value*] [`&billing_middle_name=`*value*] [`&billing_state=`*value*] [`&billing_street1=`*value*] [`&billing_street2=`*value*] [`&billing_street3=`*value*] [`&billing_zip=`*value*] [`&check_number=`*value*] [`&city=`*value*] [`&country=`*value*] [`&credit_card_month=`*value*] [`&credit_card_number=`*value*] [`&credit_card_verification_code=`*value*] [`&credit_card_year=`*value*] [`&email=`*value*] [`&gift_aid_status=`*value*] [`&gift_category_id=`*value*] [`&gift_display_name=`*value*] [`&gift_display_personal_page=`*value*] [`&middle_name=`*value*] [`&prof_suffix=`*Esq.* | *MD* | *JD* | *PhD* | *DO* | *DDS* | *DVM* | *PE*] [`&state=`*value*] [`&street1=`*value*] [`&street2=`*value*] [`&suffix=`*Sr.* | *Jr.* | *II* | *III* | *IV* | *V*] [`&team_gift=`*value*] [`&title=`*Mr.* | *Ms.* | *Mrs.* | *Miss* | *Dr.*] [`&zip=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRTeamraiserAPI?method=addGift &api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&cons_id=`*value* `&email=`*value* `&first_name=`*value* `&gift_amount=`*value* `&last_name=`*value* `&payment_type=`*cash* | *check* `&team_id=`*value* [`&anonymous=`*value*] [`&billing_city=`*value*] [`&billing_county=`*value*] [`&billing_first_name=`*value*] [`&billing_last_name=`*value*] [`&billing_middle_name=`*value*] [`&billing_state=`*value*] [`&billing_street1=`*value*] [`&billing_street2=`*value*] [`&billing_street3=`*value*] [`&billing_zip=`*value*] [`&check_number=`*value*] [`&city=`*value*] [`&confirmed=`*value*] [`&country=`*value*] [`&event_gift=`*value*] [`&fr_id=`*value*] [`&gift_aid_status=`*value*] [`&gift_category_id=`*value*] [`&gift_date=`*value*] [`&gift_display_name=`*value*] [`&gift_display_personal_page=`*value*] [`&honor_gift_message_body=`*value*] [`&honor_gift_type=`*memorial* | *honor*] [`&honoree_first_name=`*value*] [`&honoree_last_name=`*value*] [`&honoree_title=`*value*] [`&middle_name=`*value*] [`&prof_suffix=`*Esq.* | *MD* | *JD* | *PhD* | *DO* | *DDS* | *DVM* | *PE*] [`&state=`*value*] [`&street1=`*value*] [`&street2=`*value*] [`&suffix=`*Sr.* | *Jr.* | *II* | *III* | *IV* | *V*] [`&team_gift=`*value*] [`&title=`*Mr.* | *Ms.* | *Mrs.* | *Miss* | *Dr.*] [`&zip=`*value*]

Usage Notes

This method allows participants to record unconfirmed offline TeamRaiser gifts on behalf of themselves or, if enabled, their teams. It also permits online processing of credit card transactions.

This method must be invoked over a secure channel by a logged in user.

Note: This method requires an authentication token.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

first_name

Required. First name

Type xsd:string.

gift_amount

Required. Gift amount

Type xsd:string.

last_name

Required. Last name

Type xsd:string.

payment_type

Required. The payment type

Type xsd:string.

Default is cash.

Options are:

- cash
- check
- later
- credit
- ach

team_id

Required. A unique team ID

Type xsd:nonNegativeInteger.

billing_city

Optional. Billing information: City

Type xsd:string.

billing_county

Optional. Billing information: County

Type xsd:string.

billing_first_name

Optional. Billing information: First name

Type xsd:string.

billing_last_name

Optional. Billing information: Last name

Type xsd:string.

billing_middle_name

Optional. Billing information: Middle name

Type xsd:string.

billing_state

Optional. Billing information: State

Type xsd:string.

billing_street1

Optional. Billing information: Street1

Type xsd:string.

billing_street2

Optional. Billing information: Street2

Type xsd:string.

billing_street3

Optional. Billing information: Street3

Type xsd:string.

billing_zip

Optional. Billing information: Zip/Postal code

Type xsd:string.

check_number

Optional. Check number

Type xsd:string.

city

Optional. City

Type xsd:string.

country

Optional. Country

Type xsd:string.

credit_card_month

Optional. Credit card expiration month

Type xsd:string.

credit_card_number

Optional. Credit card number

Type xsd:string.

credit_card_verification_code

Optional. Credit card verification code

Type xsd:string.

credit_card_year

Optional. Credit card expiration year

Type xsd:string.

email

Optional. E-Mail address

Type xsd:string.

gift_aid_status

Optional. Gift Aid status

Type xsd:nonNegativeInteger.

Default is 0.

gift_category_id

Optional. The gift category id

Type xsd:string.

gift_display_name

Optional. Gift display name

Type xsd:string.

gift_display_personal_page

Optional. Display gift on personal page

Type xsd:boolean.

middle_name

Optional. Middle name

Type xsd:string.

prof_suffix

Optional. Professional suffix

Type xsd:string.

Default is .

Options are:

- Esq.
- MD
- JD
- PhD
- DO
- DDS
- DVM
- PE

state

Optional. State

Type xsd:string.

street1

Optional. Street1

Type xsd:string.

street2

Optional. Street2

Type xsd:string.

suffix

Optional. Suffix

Type xsd:string.

Default is .

Options are:
- Sr.
- Jr.
- II
- III
- IV
- V

team_gift

Optional. This should be set to 'REQUIRED' if the gift is a team gift.

Type xsd:boolean.

title

Optional. Title

Type xsd:string.

Default is .

Options are:
- Mr.
- Ms.
- Mrs.
- Miss
- Dr.

zip

Optional. Postal or Zip code

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

cons_id

Required. A unique constituent ID.

Type xsd:nonNegativeInteger.

email

Required. E-Mail address

Type xsd:string.

first_name

Required. First name

Type xsd:string.

gift_amount

Required. Gift amount

Type xsd:string.

last_name

Required. Last name

Type xsd:string.

payment_type

Required. The payment type

Type xsd:string.

Default is cash.

Options are:
- cash
- check

team_id

Required. A unique team ID

Type xsd:nonNegativeInteger.

anonymous

Optional. Display this gift as anonymous.

Type xsd:boolean.

billing_city

Optional. Billing information: City

Type xsd:string.

billing_county

Optional. Billing information: County

Type xsd:string.

billing_first_name

Optional. Billing information: First name

Type xsd:string.

billing_last_name

Optional. Billing information: Last name

Type xsd:string.

billing_middle_name

Optional. Billing information: Middle name

Type xsd:string.

billing_state

Optional. Billing information: State

Type xsd:string.

billing_street1

Optional. Billing information: Street1

Type xsd:string.

billing_street2

Optional. Billing information: Street2

Type xsd:string.

billing_street3

Optional. Billing information: Street3

Type xsd:string.

billing_zip

Optional. Billing information: Zip/Postal code

Type xsd:string.

check_number

Optional. Check number

Type xsd:string.

city

Optional. City

Type xsd:string.

confirmed

Optional. Whether or not the gift is confirmed.

Type xsd:boolean.

country

Optional. Country

Type xsd:string.

event_gift

Optional. This should be set to 'REQUIRED' if the gift is an event gift.

Type xsd:boolean.

fr_id

Optional. A unique Teamraiser ID

Type xsd:nonNegativeInteger.

gift_aid_status

Optional. Gift Aid status

Type xsd:nonNegativeInteger.

Default is 0.

gift_category_id

Optional. The gift category id

Type xsd:string.

gift_date

Optional. The date the gift was recorded (YYYY-MM-DD)

Type xsd:string.

gift_display_name

Optional. Gift display name

Type xsd:string.

gift_display_personal_page

Optional. Display gift on personal page

Type xsd:boolean.

honor_gift_message_body

Optional. Honor gift message body. Applies to confirmed gifts only.

Type xsd:string.

honor_gift_type

Optional. Type of honor gift. Valid values are honor or memorial. Applies to confirmed gifts only.

Type xsd:string.

Default is .

Options are:
- memorial
- honor

honoree_first_name

Optional. Honoree first name. Applies to confirmed gifts only.

Type xsd:string.

honoree_last_name

Optional. Honoree last name. Applies to confirmed gifts only.

Type xsd:string.

honoree_title

Optional. Honoree title. Applies to confirmed gifts only.

Type xsd:string.

middle_name

Optional. Middle name

Type xsd:string.

prof_suffix

Optional. Professional suffix

Type xsd:string.

Default is .

Options are:
- Esq.
- MD
- JD
- PhD
- DO
- DDS
- DVM
- PE

state

Optional. State

Type xsd:string.

street1

Optional. Street1

Type xsd:string.

street2

Optional. Street2

Type xsd:string.

suffix

Optional. Suffix

Type xsd:string.

Default is .

Options are:
- Sr.
- Jr.
- II
- III
- IV
- V

team_gift

Optional. This should be set to 'REQUIRED' if the gift is a team gift.

Type xsd:boolean.

title

Optional. Title

Type xsd:string.

Default is .

Options are:
- Mr.
- Ms.
- Mrs.
- Miss
- Dr.

zip

Optional. Postal or Zip code

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<addGiftResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <gift>
    <acknowledged>false</acknowledged>
```

```
    <checkNumber xsi:nil="true"/>
    <confirmed>true</confirmed>
    <date>2013-02-03T05:18:26.618-06:00</date>
    <email>name1@example.com</email>
    <giftAmount>2501</giftAmount>
    <giftAidStatus>1</giftAidStatus>
    <giftMessage>Sample gift message.</giftMessage>
    <id>12001</id>
    <isRecurring>false</isRecurring>
    <name>
      <first>Jo</first>
      <middle>1</middle>
      <last>Public</last>
    </name>
    <paymentType>CREDIT</paymentType>
    <recognitionName>Sample recognition name 1</recognitionName>
  </gift>
</addGiftResponse>
```

JSON response

```
{"addGiftResponse":{"gift":{"giftMessage":"Sample gift
 message.","isRecurring":"false","paymentType":"CREDIT","checkNumber":null,"date":"2013-0
 recognition name
 1","confirmed":"true","email":"name1@example.com","giftAidStatus":"1","acknowledged":"fa
{"last":"Public","middle":"1","first":"Jo"},"giftAmount":"2501"}}}
```

addTeamraiserData

Creates a Teamraiser.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	SRTeamraiserAPI	POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=addTeamraiserData`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&add_teamraiser=`*value*

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRTeamraiserAPI?method=addTeamraiserData`
`&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] `&add_teamraiser=`*value*

Usage Notes

This method accepts a list containing one or more TeamRaiser Event data blocks, up to a system-configurable maximum of 1000 Events, and asynchronously adds them to the Convio database.

Input data is passed url-encoded as the value of the add_teamraiser parameter in the following format:

```
<addTeamraiserRequest xmlns="http://convio.com/crm/v1.0">
  <baseFrId>1010</baseFrId>
  <createAsChild>true</createAsChild>
  <createAsBlueprint>false</createAsBlueprint>
  <teamraiserCreate>
```

```
    <eventStatus>2</eventStatus>
    <eventDateDay>3</eventDateDay>
    <eventDateMonth>11</eventDateMonth>
    <eventDateYear>2011</eventDateYear>
    <eventTitle>Mathathon 78705</eventTitle>
    <eventUserTitle>Mathathon Austin Texas</eventUserTitle>
    <eventLocalOfficeAttn>Chris</eventLocalOfficeAttn>
    <eventLocalOfficeStreet1>123 Main St</eventLocalOfficeStreet1>
    <eventLocalOfficeStreet2>Unit 194</eventLocalOfficeStreet2>
    <eventLocalOfficeCity>Austin</eventLocalOfficeCity>
    <eventLocalOfficeState>TX</eventLocalOfficeState>
    <eventLocalOfficeZip>78726</eventLocalOfficeZip>
    <fundraisingCampaign>1021</fundraisingCampaign>
    <fundraisingForm>4</fundraisingForm>
    <eventCategory>General</eventCategory>
    <eventInterest>Math</eventInterest>
    <prevFrId>1000</prevFrId>
    <classificationId>Generic</classificationId>
    <classificationId2>ABC123</classificationId2>
    <fiscalYear>2010</fiscalYear>
    <locale>en_US</locale>
    <publicEventType>EventType3</publicEventType>
    <eventDate>March 11th</eventDate>
    <eventLocation>Cowboys Stadium</eventLocation>
    <eventLocationStreet1>One Legends Way</eventLocationStreet1>
    <eventLocationCity>Arlington</eventLocationCity>
    <eventLocationState>TX</eventLocationState>
    <eventLocationZip>76011</eventLocationZip>
    <eventLocationCountry>United States</eventLocationCountry>
    <eventLocationCounty>Dallas County</eventLocationCounty>
    <eventLocationRegion>North Texas</eventLocationRegion>
    <eventLocationArea>Dallas Metro</eventLocationArea>
    <eventSchedule>Arrive at 11. The math begins at noon, prizes at 3pm.</
eventSchedule>
    <mapLink>www.maps.google.com/cowboysstadium</mapLink>
    <coordinatorEmail>jjones@cowboys.com</coordinatorEmail>
    <coordinatorName>Jerry Jones</coordinatorName>
    <coordinatorPhone>(512) 555-5555</coordinatorPhone>
    <sponsor1>Pepsi</sponsor1>
    <sponsor2>Frito Lay</sponsor2>
    <sponsor3>NFL</sponsor3>
    <sponsor4>Ford</sponsor4>
    <sponsor5>ESPN</sponsor5>
  </teamraiserCreate>
  <teamraiserCreate>
    <eventDateDay>3</eventDateDay>
    <eventDateMonth>11</eventDateMonth>
    <eventDateYear>2011</eventDateYear>
    <eventTitle>Mathathon 55901</eventTitle>
    <eventUserTitle>Mathathon Rochester</eventUserTitle>
  </teamraiserCreate>
  <teamraiserCreate>
    <eventDateDay>3</eventDateDay>
    <eventDateMonth>11</eventDateMonth>
    <eventDateYear>2011</eventDateYear>
    <eventTitle>Mathathon 77047</eventTitle>
    <eventUserTitle>Mathathon Houston</eventUserTitle>
  </teamraiserCreate>
</addTeamraiserRequest>
```

Within the `<addTeamraiserRequest>` structure, the `<baseFrId>` is required. It specifies a blueprint event that will be used for all events added by the call. This event will be copied and all of its properties will be

updated by the specified properties in each `<teamRaiserCreate>` element. The `<createAsChild>` and `<createAsBlueprint>` elements are optional, and default false. Their behavior is described in the following table:

createAsBlueprint	**createAsChild**	**Result**
false	false	Creates a TeamRaiser copy of the TeamRaiser event or Blueprint specified by `baseFrId`.
true	false	Creates a new Blueprint from the Blueprint specified by `baseFrId`.
false	true	Creates a child TeamRaiser Event from the Blueprint specified by `baseFrId`.
true	true	Creates a child Blueprint from the Blueprint specified by `baseFrId`.

The <addTeamraiserRequest> also contains a collection of one or more `<teamraiserCreate>` elements. Each specifies the TeamRaiser parameters that will be used to build the event. In each `<teamraiserCreate>` block, you must specify valid values for `<eventDateDay>`, `<eventDateMonth>`, `<eventDateYear>`, and `<eventTitle>` elements, and the value of `<eventTitle>` must be unique. All other elements are optional.

The elements of the `<TeamRaiserCreate>` structure are described as follows:

eventStatus
The status of the event: Unpublished = 0 (default), Accepting Registrations Only = 1, Accepting Registrations and Gifts = 2, Accepting Gifts Only = 3, Closed = 4, Archived = 8, Deleted = 9

eventTitle
The internal name of the TeamRaiser. It must be unique and is used for matching to update existing TeamRaisers.

eventUserTitle
The public title of the TeamRaiser. Also used for page customization as the 'Event Title:' field.

eventDate
The date or dates of the event in any format. Used for page customization as the 'Event Date/Time:' field, but not stored in the event record.

eventDateMonth
The month when the event occurs in numeric (MM) format. Used to set the 'Event Date:' field.

eventDateDay
The day of the month when the event occurs in numeric (DD) format. Used to set the 'Event Date:' field.

eventDateYear
The year when the event occurs in numeric (YYYY) format. Used to set the 'Event Date:' field.

eventLocation
The location of the event. Used to set the 'Location Name:' field of the event record. Also used for page customization as the 'Event Location:' field.

eventLocationStreet1
The street address of the event. Used to set the 'Street Address:' field of the event record.

eventLocationCity	The city where the event takes place. Used to set the 'City:' field of the event record. Also used for page customization as the 'Event City:' field.
eventLocationState	The state where the event takes place. Used to set the 'State:' field of the event record. Also used for page customization as the 'Event State:' field.
eventLocationZip	The ZIP code of the location where the event takes place. Used to set the 'ZIP Code:' field of the event record.
eventLocationCountry	The country of the location where the event takes place.
eventLocationCounty	The county of the location where the event takes place.
eventLocationRegion	The region of the location where the event takes place.
eventLocationArea	The area of the location where the event takes place.
eventLocalOfficeAttn	Indicates to whom mail to the local office should be addressed.
eventLocalOfficeStreet1	First street line of the local office mailing address.
eventLocalOfficeStreet2	Second street line of the local office mailing address.
eventLocalOfficeCity	City for the local office mailing address.
eventLocalOfficeState	State for the local office mailing address.
eventLocalOfficeZip	Zip code for the local office mailing address.
eventSchedule	The schedule of the TeamRaiser. Used in the page customization as the 'Event Schedule:' field.
fundraisingCampaign	A fundraising donation campaign to associate with the event.
fundraisingForm	A fundraising donation form to associate with the event. If you specify a Donations2 campaign, you must also specify a donation form. A Donation Campaign must be specified to update the donation form.
eventCategory	The security category of the event.
eventInterest	An interest category to associate with the event.
mapLink	A web address (URL) of a map to the TeamRaiser. Used in the page customization as the 'Link to Event Directions:' field.
coordinatorName	The name of the TeamRaiser coordinator. Used in the page customization as the 'Event Coordinator Name:' field.
coordinatorPhone	The phone number of the TeamRaiser coordinator. Used in the page customization as the 'Event Coordinator Phone:' field.

coordinatorEmail	The email address of the TeamRaiser coordinator. Used in the page customization as the 'Event Coordinator Email:' field.
sponsor1	The first name or image filename for a sponsor of the TeamRaiser. Used in the page customization as the 'Sponsor name or image filename (1):' field.
sponsor2	The second name or image filename for a sponsor of the TeamRaiser. Used in the page customization as the 'Sponsor name or image filename (2):' field.
sponsor3	The third name or image filename for a sponsor of the TeamRaiser. Used in the page customization as the 'Sponsor name or image filename (3):' field.
sponsor4	The fourth name or image filename for a sponsor of the TeamRaiser. Used in the page customization as the 'Sponsor name or image filename (4):' field.
sponsor5	The fifth name or image filename for a sponsor of the TeamRaiser. Used in the page customization as the 'Sponsor name or image filename (5):' field.
prevFrId	The id or name of the old TeamRaiser event we wish to associate this new TeamRaiser with.
classificationId	The ID or the name of the type for this TeamRaiser, also called the Primary Event Type. (Note that this value may be blank.)
classificationId2	A second name of the type for this TeamRaiser. (Note that this value may be blank, Name only no ID will be accepted.)
fiscalYear	The fiscal year for this TeamRaiser.
locale	The locale of this TeamRaiser event.
publicEventType	The ID or name of the public event type for this TeamRaiser. (Note that this value may be blank.)

If the call is successful, it returns a unique job ID for the batch.

Prerequisites

API Administrator accounts that access this API must be assigned TeamRaiser API User permission under the TeamRaiser Management permission type.

This API requires that the TeamRaiser Blueprinting feature be enabled for your Convio site.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

add_teamraiser

Required. XML document specifying the TeamRaiser events to add.

Type crm:addTeamraiserList.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

add_teamraiser

Required. XML document specifying the TeamRaiser events to add.

Type crm:addTeamraiserList.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<addTeamraiserResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <jobId>1359890305306</jobId>
</addTeamraiserResponse>
```

JSON response

```
{"addTeamraiserResponse":{"jobId":"1359890305306"}}
```

addTeamraiserEventManager

Assigns an existing constituent an Event Manager role.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRTeamraiserAPI	POST	No.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRTeamraiserAPI?method=addTeamraiserEventManager&api_key=`*value*`&login_name=`*value*`&login_password=`*value*`&v=`*value*`[&center_id=`*value*`][&response_format=`*xml*`|`*json*`][&source=`*value*`][&sub_source=`*value*`][&suppress_response_codes=`*value*`]&cons_id=`*value*`&fr_id=`*value*`&role_id=`*value*

Usage Notes

This method adds the specified constituent as an event manager for a TeamRaiser event and grants them the privileges defined by the specified event manager role.

If the constituent is already an event manager for the specified event, their role is updated to match the role specified.

This Server API is accessible only from white-listed servers. API Administrator accounts that access this API must be assigned TeamRaiser API User permission under the TeamRaiser Management permission type.

See also: *listTeamraiserEventManagers*, *removeTeamraiserEventManager* and *getTeamraiserEventManagerRoles*.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Required. A unique constituent ID.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

role_id

Required. Event Manager Role ID

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<addTeamraiserEventManagerResponse xmlns="http://convio.com/crm/v1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd">true</success>
</addTeamraiserEventManagerResponse>
```

JSON response

```
{"addTeamraiserEventManagerResponse":{"success":
{"content":"true","xsi:schemaLocation":"http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd"}}}
```

deleteDraft

Deletes of an existing draft

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=deleteDraft`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* `&message_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

message_id

Required. Message ID

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<deleteDraftResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <messageId>1023</messageId>
</deleteDraftResponse>
```

JSON response

```
{"deleteDraftResponse":{"messageId":"1023"}}
```

deleteGift

Provides a method for deleting offline unconfirmed gifts.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=deleteGift`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* `&gift_id=`*value* [`&team_gift=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

gift_id

Required. Unique gift ID

Type xsd:nonNegativeInteger.

team_gift

Optional. This should be set to 'REQUIRED' if the gift is a team gift.

Type xsd:boolean.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<deleteGiftResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <giftId>1</giftId>
</deleteGiftResponse>
```

JSON response

```
{"deleteGiftResponse":{"giftId":"1"}}
```

deleteSentMessage

Deletes a sent message from the sent message list.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=deleteSentMessage` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&message_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

message_id

Required. Message ID

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<deleteSentMessageResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <messageId>1022</messageId>
</deleteSentMessageResponse>
```

JSON response

```
{"deleteSentMessageResponse":{"messageId":"1022"}}
```

deleteTeamraiserAddressBookContacts

Deletes one or more contacts from the owner's address book, including contacts shared by team members. Shared contacts are not deleted from team members Address Books, only from the owner's address book.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=deleteTeamraiserAddressBookContacts &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&contact_ids=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Similar to the Address Book API's *deleteAddressBookContacts* method, this method allows you to remove contacts from a constituent's Address Book.

The primary difference between this method and *deleteAddressBookContacts* is that this method requires the fr_id of the event the participant is currently working in, and is capable of removing contacts shared by team members.

When removing a contact that is shared by team members (such as a Team donor, or another team member) the removal affects only the currently logged in participant - the rest of the team will continue to see the contact in their address book (depending on site configuration).

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

contact_ids

Optional. Comma-separated list of contact IDs to delete.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<deleteTeamraiserAddressBookContactsResponse xsi:schemaLocation="http://
convio.com/crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd"
 xmlns="http://convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <contactId>101</contactId>
  <contactId>102</contactId>
  <contactId>103</contactId>
  <errors>false</errors>
</deleteTeamraiserAddressBookContactsResponse>
```

JSON response

```
{"deleteTeamraiserAddressBookContactsResponse":{"errors":"false","contactId":
["101","102","103"]}}
```

getCampaignByNameData

Provides a TeamRaiser event by name.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getCampaignByNameData &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&name=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

name

Required. The name of the TeamRaiser event.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getRegisteredTeamraisersResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <teamraiser>
    <id>12001</id>
    <name>Sample Event Title 1</name>
    <description>Sample Event Description 1</description>
    <event_date>2013-02-03T05:18:25.441-06:00</event_date>
    <accepting_registrations>true</accepting_registrations>
    <accepting_donations>true</accepting_donations>
    <location_name>Sample Location 1</location_name>
    <street_address>1234 Fake St.</street_address>
    <city>Austin</city>
    <state>TX</state>
    <zip>78758</zip>
    <county>Travis</county>
    <country>United States</country>
    <region>Hill Country</region>
    <area>North Austin</area>
    <mailingAddrAttn>J. Doe</mailingAddrAttn>
    <mail_street1>1234 Fake Ave.</mail_street1>
    <mail_street2>Suite 1000</mail_street2>
    <mail_city>Austin</mail_city>
    <mail_state>TX</mail_state>
    <mail_zip>78758</mail_zip>
    <status>1</status>
    <greeting_url>http://example.com/greetingUrl</greeting_url>
    <find_url>http://example.com/findTeamOrParticipantUrl</find_url>
    <donate_event_url>http://example.com/donateUrl</donate_event_url>
    <reg_indiv_url>http://example.com/registerIndividualUrl</reg_indiv_url>
    <reg_new_team_url>http://example.com/formANewTeamUrl</reg_new_team_url>
    <reg_join_team_url>http://example.com/joinTeamUrl</reg_join_team_url>
    <public_event_type_name>Central Texas</public_event_type_name>
    <ecommerce_event_url>http://example.com/eCommerceEventUrl</
ecommerce_event_url>
  </teamraiser>
</getRegisteredTeamraisersResponse>
```

JSON response

```
{"getRegisteredTeamraisersResponse":{"teamraiser":{"region":"Hill
 Country","state":"TX","city":"Austin","id":"12001","reg_new_team_url":"http://
example.com/formANewTeamUrl","mail_state":"TX","area":"North
 Austin","description":"Sample Event Description 1","name":"Sample
 Event Title 1","mail_city":"Austin","reg_indiv_url":"http://
example.com/registerIndividualUrl","ecommerce_event_url":"http://
example.com/eCommerceEventUrl","public_event_type_name":"Central
 Texas","zip":"78758","reg_join_team_url":"http://
example.com/joinTeamUrl","location_name":"Sample
 Location 1","status":"1","find_url":"http://example.com/
```

```
findTeamOrParticipantUrl","country":"United
 States","accepting_donations":"true","greeting_url":"http://
example.com/greetingUrl","donate_event_url":"http://example.com/
donateUrl","county":"Travis","event_date":"2013-02-03T05:18:25.442-06:00","mail_zip":"787
 Doe","street_address":"1234 Fake
 St.","accepting_registrations":"true","mail_street1":"1234 Fake
 Ave.","mail_street2":"Suite 1000"}}}
```

getCaptainsMessage

Provides the message a team captain writes to the team and whether or not a participant can edit that message.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getCaptainsMessage &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getCaptainsMessageResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <message>Sample captains message.</message>
  <editable>false</editable>
</getCaptainsMessageResponse>
```

JSON response

```
{"getCaptainsMessageResponse":{"message":"Sample captains
 message.","editable":"false"}}
```

getCompaniesByInfo

Provides detailed information about a company.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getCompaniesByInfo`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&company_id=`*value*] [`&company_name=`*value*] [`&event_type=`*value*] [`&fr_id=`*value*] [`&include_cross_event=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*company_name* | *total* | *fr_id* | *company_id* | *coordinator_cons_id* | *goal* | *national* | *regional* | *classification_name*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*company_name* | *total* | *fr_id* | *company_id* | *coordinator_cons_id* | *goal* | *national* | *regional* | *classification_name*]

Usage Notes

This method requires that you provide at least one of the following parameters:

- fr_id
- company_id
- company_name
- event_type

Note: Only active, participating companies are returned. Prospect or Sponsor-only companies will not be returned.

Note: When searching by company_name or event_type, you must supply a minimum number of characters to search on. Default is 3, but this may be set by the Site Data Parameter F2F_PART_FINDER_MIN_CRITERIA.

Amounts and Counts

The Amount field will only be included in the response in the following conditions:

- The request included the fr_id parameter or the company_id parameter and the company is a Local Company.
- The request included the fr_id parameter, the include_cross_event parameter set to true, and the company is a National or Regional Team.

The way amounts are displayed in status indicators may vary from event to event based on the event configuration. A calculated amount for cross-event teams respecting that configuration is not available at this time.

For Cross-Event Teams, if you specify the fr_id the teamCount and participantCount fields in the response will be limited to that teams and participants associated to that company in the specified TeamRaiser event. If you do not specify the fr_id, the teamCount and participantCount will be the count of all teams and participants associated to that National or Regional Team across all the TeamRaiser events that National or Regional Team is associated with.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

company_id

Optional. The id of the company.

Type xsd:nonNegativeInteger.

company_name

Optional. The public name of the company.

Type xsd:string.

event_type

Optional. The name of the Primary Event Type to limit search results to. Omit this parameter to search on all Primary Event Types. Defaults to null.

Type xsd:string.

fr_id

Optional. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

include_cross_event

Optional. Tells us if we want to make sure to include Cross-Event teams in our queries

Type xsd:boolean.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

Options are:

- company_name
- total
- fr_id
- company_id
- coordinator_cons_id
- goal
- national
- regional
- classification_name

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

Options are:

- company_name
- total
- fr_id
- company_id
- coordinator_cons_id
- goal
- national
- regional
- classification_name

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getCompaniesResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>3</totalNumberResults>
  <company>
    <eventId>1000</eventId>
    <companyId>2000</companyId>
    <companyName>sampleCompany0</companyName>
    <eventType>eventType0</eventType>
    <coordinatorId>1000</coordinatorId>
    <goal>4000</goal>
    <amountRaised>5000</amountRaised>
    <companyURL>www.blackbaud.com0</companyURL>
    <companyType>National</companyType>
    <teamCount>731</teamCount>
    <participantCount>814</participantCount>
  </company>
  <company>
    <eventId>1001</eventId>
    <companyId>2001</companyId>
    <companyName>sampleCompany1</companyName>
    <eventType>eventType1</eventType>
    <coordinatorId>1001</coordinatorId>
    <goal>4001</goal>
    <amountRaised>5001</amountRaised>
    <companyURL>www.blackbaud.com1</companyURL>
    <companyType>National</companyType>
    <teamCount>101</teamCount>
    <participantCount>579</participantCount>
  </company>
  <company>
    <eventId>1002</eventId>
    <companyId>2002</companyId>
    <companyName>sampleCompany2</companyName>
    <eventType>eventType2</eventType>
    <coordinatorId>1002</coordinatorId>
    <goal>4002</goal>
    <amountRaised>5002</amountRaised>
```

```
    <companyURL>www.blackbaud.com2</companyURL>
    <companyType>National</companyType>
    <teamCount>829</teamCount>
    <participantCount>240</participantCount>
  </company>
</getCompaniesResponse>
```

JSON response

```
{"getCompaniesResponse":{"company":
[{"eventId":"1000","companyURL":"www.blackbaud.com0","amountRaised":"5000","participantCo
{"eventId":"1001","companyURL":"www.blackbaud.com1","amountRaised":"5001","participantCou
{"eventId":"1002","companyURL":"www.blackbaud.com2","amountRaised":"5002","participantCou
```

Examples

To retrieve the local companies associated with the TeamRaiser with the fr_id=1000:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=api_key&v=1.0&method=getCompaniesByInfo&fr_id=1000
```

To retrieve the local companies and and cross-event teams associated with the TeamRaiser with the fr_id=1000:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=api_key&v=1.0&method=getCompaniesByInfo&fr_id=1000&include_cross_event=true
```

If company_id=1000 is the id of a National Team, this call will **not** return you the details of the company, but will return a count of the companies that match the criteria. If company_id=1000 is a local team, it will return the details of the company.

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=api_key&v=1.0&method=getCompaniesByInfo&company_id=1000
```

To fix the call above, where company_id=1000 is the id of a National Team, include the include_cross_event=true in order to retrieve the details for that company.

Remember however, that this example will not include the Amount field, since we do not provide the fr_id, and the teamCount and participantCount will include all teams and participants across all events:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=api_key&v=1.0&method=getCompaniesByInfo&company_id=1000&include_cross_event=true
```

Modifying the above call to provide the fr_id will retrieve the amount that company has raised in that TeamRaiser and will change the teamCount and participantCount to only count teams and participants within that TeamRaiser.

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=api_key&v=1.0&method=getCompaniesByInfo&fr_id=1000&company_id=1000&include_cross_
```

To search for companies where "National" appears in the company name:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=api_key&v=1.0&method=getCompaniesByInfo&company_name=National
```

Note: Note that if you have 2 Cross-Event Teams with "National" in their name and one Local company with "National" in the name, this will return a count of 3 but only include the details of the Local Company.

Fixing the above and searching for companies where "National" appears in the company name and retrieve the details of all 3 companies:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=api_key&v=1.0&method=getCompaniesByInfo&company_name=National&include_cross_event
```

Searching for Companies in the event type of "Walkfest":

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=api_key&v=1.0&method=getCompaniesByInfo&event_type=Walkfest
```

Note: Since local companies cannot be associated to event types, this will return a count of the cross-event teams associated to "Walkfest", but will not include the details of those companies.

Fixing the above to return the company details for Cross-Event Teams in the event type of "Walkfest":

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=api_key&v=1.0&method=getCompaniesByInfo&event_type=Walkfest&include_cross_event=t
```

getCompanyDonationByDay

Provides data used to build daily progress charts.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getCompanyDonationByDay &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getCompanyDonationByDayResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <donationDataByDay>
    <date>2013-02-02-06:00</date>
```

```
    <amount>4</amount>
  </donationDataByDay>
  <donationDataByDay>
    <date>2013-02-03-06:00</date>
    <amount>8</amount>
  </donationDataByDay>
</getCompanyDonationByDayResponse>
```

JSON response

```
{"getCompanyDonationByDayResponse":{"donationDataByDay":
[{"amount":"4","date":"2013-02-02-06:00"},
{"amount":"8","date":"2013-02-03-06:00"}]}}
```

getCompanyList

Returns a list of all available existing companies associated with a TeamRaiser ID.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getCompanyList`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&include_all_companies=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

include_all_companies

Optional. Specifies whether local companies that are not available for team selection should be included in the response.

Type xsd:boolean.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getCompanyListResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <companyItem>
    <companyId>0</companyId>
    <companyName>example_company0</companyName>
  </companyItem>
  <companyItem>
```

```
    <companyId>1</companyId>
    <companyName>example_company1</companyName>
  </companyItem>
</getCompanyListResponse>
```

JSON response

```
{"getCompanyListResponse":{"companyItem":
[{"companyId":"0","companyName":"example_company0"},
{"companyId":"1","companyName":"example_company1"}]}}
```

getCompanyPageInfo

Provides information used to display or edit a company page.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getCompanyPageInfo &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getCompanyPageResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <companyPage>
    <pageTitle>Sample Page Title</pageTitle>
```

```
    <richTextEnabled>true</richTextEnabled>
    <richText>&lt;p>Sample HTML.&lt;/p></richText>
  </companyPage>
</getCompanyPageResponse>
```

JSON response

```
{"getCompanyPageResponse":{"companyPage":{"pageTitle":"Sample Page
 Title","richText":"<p>Sample HTML.<\/p>","richTextEnabled":"true"}}}
```

getCompanyPagePreview

Retrieves a preview of the company page.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getCompanyPagePreview &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&headline1=`*value*] [`&headline2=`*value*] [`&headline3=`*value*] [`&page_title=`*value*] [`&rich_text=`*value*] [`&status1=`*value*] [`&status2=`*value*] [`&status3=`*value*] [`&status4=`*value*] [`&status5=`*value*] [`&text1=`*value*] [`&text2=`*value*] [`&text3=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method requires authentication, and must include the company_id. Only the logged-in constituent who is identified as the Company Coordinator in the Company record has permission to execute this method.

The Company Coordinator is defined by the Convio system administrator when the Company record is created.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

headline1

Optional. The first headline on the page.

Type xsd:string.

headline2

Optional. The second headline on the page.

Type xsd:string.

headline3

Optional. The third headline on the page.

Type xsd:string.

page_title

Optional. The title of the page

rich_text

Optional. The rich text area of the page.

Type xsd:string.

status1

Optional. Status Indicator 1

Type xsd:string.

status2

Optional. Status Indicator 2

Type xsd:string.

status3

Optional. Status Indicator 3

Type xsd:string.

status4

Optional. Status Indicator 4

Type xsd:string.

status5

Optional. Status Indicator 5

Type xsd:string.

text1

Optional. The first text area on the page.

Type xsd:string.

text2

Optional. The second text area on the page.

Type xsd:string.

text3

Optional. The third text area on the page.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getCompanyPagePreviewResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <previewUrl>http://www.example.com/preview.html</previewUrl>
</getCompanyPagePreviewResponse>
```

JSON response

```
{"getCompanyPagePreviewResponse":{"previewUrl":"http://www.example.com/
preview.html"}}
```

getCompanyPhoto

Retrieves the company photo for a team.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getCompanyPhoto` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&company_id=`*value* `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method requires authentication, and must include the company_id. Only the logged-in constituent who is identified as the Company Coordinator in the Company record has permission to execute this method.

The Company Coordinator is defined by the Convio system administrator when the Company record is created.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

company_id

Required. The unique ID of the company.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getCompanyPhotoResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <photoItem>
    <id>1</id>
    <thumbnailUrl>&lt;url></thumbnailUrl>
    <customUrl>&lt;url></customUrl>
    <originalUrl>&lt;url></originalUrl>
  </photoItem>
</getCompanyPhotoResponse>
```

JSON response

```
{"getCompanyPhotoResponse":{"photoItem":
{"id":"1","originalUrl":"<url>","thumbnailUrl":"<url>","customUrl":"<url>"}}}
```

getCompanyShortcut

Provides information about a company's shortcut URL.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getCompanyShortcut &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&company_id=`*value* `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method requires authentication, and must include the company_id. Only the logged-in constituent who is identified as the Company Coordinator in the Company record has permission to execute this method.

The Company Coordinator is defined by the Convio system administrator when the Company record is created.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

company_id

Required. The unique ID of the company.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getCompanyShortcutResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <shortcutItem>
    <defaultUrl>http://example.com/personalpageurl</defaultUrl>
    <hint>Sample hint text.</hint>
    <prefix>http://example.com/goto/</prefix>
    <text>sampleuser</text>
    <url>http://example.com/goto/sampleuser</url>
  </shortcutItem>
</getCompanyShortcutResponse>
```

JSON response

```
{"getCompanyShortcutResponse":{"shortcutItem":
{"text":"sampleuser","prefix":"http://example.com/goto/","hint":"Sample hint
 text.","defaultUrl":"http://example.com/personalpageurl","url":"http://
example.com/goto/sampleuser"}}}
```

getCompanyTeams

Returns the company teams for a given company.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getCompanyTeams`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [

`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&company_id=`*value*] [`&include_download_url=`*value*] [`&team_id=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

company_id

Optional. The unique ID of the company.

Type xsd:nonNegativeInteger.

include_download_url

Optional. Specifies whether a url to download the data should be included in the response.

Type xsd:boolean.

team_id

Optional. An optional team ID. If this value is set and company_id is not included, the company associated with this team will be used.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getCompanyTeamsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <companyTeam>
    <amount>1001</amount>
    <teamId>1</teamId>
    <teamName>example team 1</teamName>
    <numGifts>1</numGifts>
  </companyTeam>
  <companyTeam>
    <amount>1002</amount>
    <teamId>2</teamId>
```

```
    <teamName>example team 2</teamName>
    <numGifts>2</numGifts>
  </companyTeam>
  <companyTeam>
    <amount>1003</amount>
    <teamId>3</teamId>
    <teamName>example team 3</teamName>
    <numGifts>3</numGifts>
  </companyTeam>
</getCompanyTeamsResponse>
```

JSON response

```
{"getCompanyTeamsResponse":{"companyTeam":[{"teamName":"example team
 1","amount":"1001","numGifts":"1","teamId":"1"},{"teamName":"example team
 2","amount":"1002","numGifts":"2","teamId":"2"},{"teamName":"example team
 3","amount":"1003","numGifts":"3","teamId":"3"}]}}
```

getContactActivity

Provides a list of recent activity for display in the participant center.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getContactActivity` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&contact_id=`*value* `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

contact_id

Required. A unique ID to identify an address book contact.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getContactActivityResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <contactActivityRecord>
    <activity>Sample activity 01</activity>
    <body>Body text here.</body>
    <subject>Subject here</subject>
    <type>DONATION</type>
    <date>January 1</date>
  </contactActivityRecord>
  <contactActivityRecord>
    <activity>Sample activity 11</activity>
    <body>Body text here.</body>
    <subject>Subject here</subject>
    <type>MESSAGE</type>
    <date>January 1</date>
  </contactActivityRecord>
  <contactActivityRecord>
    <activity>Sample activity 21</activity>
    <body>Body text here.</body>
    <subject>Subject here</subject>
    <type>RECRUIT</type>
    <date>January 1</date>
  </contactActivityRecord>
  <contactActivityRecord>
    <activity>Sample activity 31</activity>
    <body>Body text here.</body>
    <subject>Subject here</subject>
    <type>DONATION</type>
    <date>January 1</date>
  </contactActivityRecord>
  <contactActivityRecord>
    <activity>Sample activity 41</activity>
    <body>Body text here.</body>
    <subject>Subject here</subject>
    <type>MESSAGE</type>
    <date>January 1</date>
  </contactActivityRecord>
  <contactActivityRecord>
    <activity>Sample activity 51</activity>
    <body>Body text here.</body>
    <subject>Subject here</subject>
    <type>RECRUIT</type>
    <date>January 1</date>
  </contactActivityRecord>
  <contactActivityRecord>
    <activity>Sample activity 61</activity>
    <body>Body text here.</body>
    <subject>Subject here</subject>
    <type>DONATION</type>
    <date>January 1</date>
  </contactActivityRecord>
  <contactActivityRecord>
```

```
    <activity>Sample activity 71</activity>
    <body>Body text here.</body>
    <subject>Subject here</subject>
    <type>MESSAGE</type>
    <date>January 1</date>
  </contactActivityRecord>
  <contactActivityRecord>
    <activity>Sample activity 81</activity>
    <body>Body text here.</body>
    <subject>Subject here</subject>
    <type>RECRUIT</type>
    <date>January 1</date>
  </contactActivityRecord>
  <contactActivityRecord>
    <activity>Sample activity 91</activity>
    <body>Body text here.</body>
    <subject>Subject here</subject>
    <type>DONATION</type>
    <date>January 1</date>
  </contactActivityRecord>
</getContactActivityResponse>
```

JSON response


```
{"getContactActivityResponse":{"contactActivityRecord":
[{"body":"Body text here.","subject":"Subject here","date":"January
 1","type":"DONATION","activity":"Sample activity
 01"},{"body":"Body text here.","subject":"Subject
 here","date":"January 1","type":"MESSAGE","activity":"Sample
 activity 11"},{"body":"Body text here.","subject":"Subject
 here","date":"January 1","type":"RECRUIT","activity":"Sample
 activity 21"},{"body":"Body text here.","subject":"Subject
 here","date":"January 1","type":"DONATION","activity":"Sample
 activity 31"},{"body":"Body text here.","subject":"Subject
 here","date":"January 1","type":"MESSAGE","activity":"Sample
 activity 41"},{"body":"Body text here.","subject":"Subject
 here","date":"January 1","type":"RECRUIT","activity":"Sample
 activity 51"},{"body":"Body text here.","subject":"Subject
 here","date":"January 1","type":"DONATION","activity":"Sample
 activity 61"},{"body":"Body text here.","subject":"Subject
 here","date":"January 1","type":"MESSAGE","activity":"Sample
 activity 71"},{"body":"Body text here.","subject":"Subject
 here","date":"January 1","type":"RECRUIT","activity":"Sample activity
 81"},{"body":"Body text here.","subject":"Subject here","date":"January
 1","type":"DONATION","activity":"Sample activity 91"}]}}
```


getDraft

Provides detailed information about a suggested message.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getDraft`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [
`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*

] [&sub_source=*value*] [&success_redirect=*value*] [&suppress_response_codes=*value*]
&fr_id=*value* &message_id=*value* [&auth=*value*] [&sso_auth_token=*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

message_id

Required. Message ID

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getDraftResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <messageInfo>
    <messageId>123</messageId>
    <subject>Sample subject</subject>
    <messageBody>Sample message body.</messageBody>
    <messageName>Sample Name</messageName>
  </messageInfo>
</getDraftResponse>
```

JSON response

```
{"getDraftResponse":{"messageInfo":{"messageBody":"Sample message
 body.","subject":"Sample subject","messageId":"123","messageName":"Sample
 Name"}}}
```

getDrafts

Provides a list of message drafts for a participant.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getDrafts`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getDraftsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>1</totalNumberResults>
  <messageItem>
    <date>1/1/09</date>
    <message_id>1001</message_id>
    <recipients>name1@example.com</recipients>
    <subject>example subject</subject>
  </messageItem>
</getDraftsResponse>
```

JSON response

```
{"getDraftsResponse":{"totalNumberResults":"1","messageItem":
{"recipients":"name1@example.com","subject":"example
 subject","message_id":"1001","date":"1/1/09"}}}
```

getEventDataParameter

Provides the value of a TeamRaiser event data parameter (EDP).

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getEventDataParameter &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&edp_name=`*value*] [`&edp_type=`*value*]

Usage Notes

Valid values for edp_name are:

- F2F_ACCEPT_OFFLINE_SERVER_API
- F2F_ACCEPT_OFFLINE

- F2F_ADMIN_NEWS_FEEDS_ENABLED
- F2F_ALLOW_EVENT_MANAGEMENT
- F2F_ALLOW_OFFLINE_PART_TYPES
- F2F_ALLOW_SHOP_FOR_PREMIUMS
- F2F_BLUEPRINT_LOCKING_ENABLED
- F2F_BLUEPRINT_SHOW_ALL_DEBUG_MODE
- F2F_CENTER_TAF_BOUNCE_NOTIFICATION_ENABLED
- F2F_CENTER_TAF_PERSONALIZED_SALUTATION_ENABLED
- F2F_CHART_DATA_API_DAYS_ALLOWED
- F2F_COMPANY_FUNDRAISING_GOALS
- F2F_COMPONENT_STYLING_ENABLED
- F2F_CROSS_JVM_CACHE_ENABLED
- F2F_CUSTOMIZED_URL_IS_PERMANENT
- F2F_DEFAULT_STATIONERY
- F2F_DISABLE_CLASSIC_PART_CENTER
- F2F_DISABLE_SELF_DONATION_RECEIPTS
- F2F_DISPLAY_LIST_DONATE_BUTTON
- F2F_DISPLAY_LIST_ECOMMERCE_BUTTON
- F2F_DONOR_HONOR_ROLL_BY_ALPHA
- F2F_ECOMMERCE_ALLOW_ENABLED
- F2F_ECOMMERCE_CREDIT_PARTICIPANT
- F2F_ECOMMERCE_CREDIT_TEAM
- F2F_ECOMMERCE_ENABLED
- F2F_ECOMMERCE_PRODUCT_ID
- F2F_ECOMMERCE_STORE_ID
- F2F_EM_CONTACT_SEARCH_MAX_RESULTS
- F2F_EM_SHOW_UNCONFIRMED_GIFTS_IN_HISTORY
- F2F_ENABLE_ACHIEVEMENT_BADGES
- F2F_ENFORCE_UNIQUE_EMAIL
- F2F_ENFORCE_UNIQUE_NAME_AND_EMAIL
- F2F_EVENT_CREATION_EMAIL
- F2F_FB_CONNECT_PUBLISH_FEED_IMG_SRC
- F2F_FB_CONNECT_PUBLISH_FEED_IMG_URL
- F2F_HTML_SUGGESTED_MESSAGES_ALLOW_IMAGES
- F2F_HTML_SUGGESTED_MESSAGES_ALLOW_LINKS
- F2F_HTML_SUGGESTED_MESSAGES_ENABLED
- F2F_INCLUDE_UNCONFIRMED
- F2F_MILESTONES_ENABLED
- F2F_MILESTONES_NUM_ALLOWED
- F2F_MULTIPLE_CAPTAINS_ENABLED
- F2F_OFFLINE_GIFT_GIFT_AID_DESC
- F2F_OFFLINE_GIFT_GIFT_AID
- F2F_OFFLINE_GIFT_NOTIFICATION_DISABLED
- F2F_OFFLINE_PMT_TYPES
- F2F_OFFLINE_TEAM_GIFTS_ALLOW_TYPE
- F2F_PART_CENTER_COOKIE_MAX_AGE
- F2F_PART_CENTER_EMAIL_DAYS
- F2F_PART_CENTER_INCLUDE_SECOND_PHOTO
- F2F_PART_CENTER_LOGOUT_REDIRECT
- F2F_PART_CENTER_MIN_AB_SIZE
- F2F_PART_CENTER_OVERRIDE_ABOOK_URL
- F2F_PART_CENTER_OVERRIDE_EMAIL_URL
- F2F_PART_CENTER_OVERRIDE_GIFT_URL
- F2F_PART_CENTER_OVERRIDE_PEDITOR_URL
- F2F_PART_CENTER_OVERRIDE_PROGRESS_URL
- F2F_PART_CENTER_OVERRIDE_SETTINGS

- F2F_PART_CENTER_OVERRIDE_TENTING_URL
- F2F_PART_CENTER_OVERRIDE_TPROGRESS_URL
- F2F_PART_CENTER_OVERRIDE_URL
- F2F_PART_CENTER_PERSONAL_PAGE_DAYS
- F2F_PART_CENTER_PROGRESS_IMAGE_ID
- F2F_PART_CENTER_SHARE_ACTION
- F2F_PART_CENTER_SHARE_ID
- F2F_PART_CENTER_SHARE_MESSAGE
- F2F_PART_CENTER_SHARE_TITLE
- F2F_PART_CENTER_UPDATE_GOAL_PERCENT
- F2F_PART_CENTER_USE_YUI_STORAGE
- F2F_PART_EDIT_REG_RESPONSES
- F2F_PARTICIPANT_ALLOW_CARD_PROCESS
- F2F_PARTICIPANT_EMAILS_TRACKABLE
- F2F_PERSONAL_BLOGS_ENABLED
- F2F_PERSONAL_PRECALCULATION_ENABLED
- F2F_PRECALCULATION_ENABLED
- F2F_PRECALCULATION_INTERVAL
- F2F_PRECALCULATION_NAME_UPDATE_ENABLED
- F2F_PRECALCULATION_NAME_UPDATE_INTERVAL
- F2F_PRECALCULATION_NAME_UPDATE_TIME
- F2F_PUBLIC_EVENT_TYPES_ENABLED
- F2F_REASSIGN_RACE_NUMBER
- F2F_RECENT_ACTIVITY_SIZE
- F2F_RECRUITING_GOALS
- F2F_REG_ANONYMOUS_CHECKBOX
- F2F_REG_INTERACTION_SHOW_SNAPSHOT
- F2F_REG_PUBLIC_CHECKBOX
- F2F_RESTRICT_STATIONERY_TO_EVENT
- F2F_RETURNING_TEAMS_AB_VISIBILITY
- F2F_RETURNING_TEAMS_COPY_PREVIOUS_COMPANY
- F2F_RETURNING_TEAMS_COPY_PREVIOUS_DIVISION
- F2F_RETURNING_TEAMS_COPY_PREVIOUS_GOAL
- F2F_RETURNING_TEAMS_ENABLED
- F2F_RETURNING_TEAMS_KEEP_PREVIOUS_ROLES
- F2F_RETURNING_TEAMS_REFORMER
- F2F_RETURNING_TEAMS_USE_PREVIOUS_TEAM_PAGE
- F2F_SCREENNAME_MAX_LENGTH
- F2F_SHOW_GIFT_CATEGORIES_FOR_PARTICIPANT_GIFTS
- F2F_STATUS_INCLUDE_SECONDARY_REG_FEES
- F2F_STATUS_THERMOMETER_PROGRESS_IMAGE
- F2F_TEAM_AB_VISIBILITY
- F2F_TEAM_MILESTONES_ENABLED
- F2F_TEAM_MILESTONES_NUM_ALLOWED
- F2F_TEAM_ROSTER_MAX_INCLUDE_AMOUNT_SIZE
- F2F_TEAMRAISER_LIST_COMPONENT_MAX_SIZE
- F2F_UPDATE_CONSTITUENT_FROM_SECONDARY_REGISTRANT
- F2F_UPLOAD_REG_ALLOW_TRANSACTIONS
- F2F_USER_CAN_CHOOSE_LAYOUT
- F2F_USER_CAN_CHOOSE_STATUS_IND
- F2F_USER_CAN_CHOOSE_STYLE
- F2F_USER_RICH_TEXT_PAGES_ALLOW_IMAGES
- F2F_USER_RICH_TEXT_PAGES_ALLOW_LINKS
- F2F_USER_RICH_TEXT_PAGES_ENABLED

Valid values edp_type are:

- Boolean

- Integer
- String

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

edp_name

Optional. The name of the requested Event Data Parameter (EDP).

edp_type

Optional. This is the way the EDP will be represented. Values are: string, integer, and boolean

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getEventDataParameterResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <stringValue>Sample Value</stringValue>
</getEventDataParameterResponse>
```

JSON response

```
{"getEventDataParameterResponse":{"stringValue":"Sample Value"}}
```

getFlowStep

Provides state information for external applications.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getFlowStep`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getFlowStepResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <flowStep>3</flowStep>
</getFlowStepResponse>
```

JSON response

```
{"getFlowStepResponse":{"flowStep":"3"}}
```

getFundraisingResults

Provides information about participant fundraising progress.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getFundraisingResults &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&cons_id=`*value* `&fr_id=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Required. A unique constituent ID.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getFundraisingResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <fundraisingRecord>
    <amountRaised>7500</amountRaised>
    <minimumGoal>25000</minimumGoal>
    <offlineConfirmedAmount>250</offlineConfirmedAmount>
    <offlineConfirmedDonationCount>1</offlineConfirmedDonationCount>
    <offlineTotalAmount>1000</offlineTotalAmount>
    <offlineTotalDonationCount>3</offlineTotalDonationCount>
    <offlineUnconfirmedAmount>750</offlineUnconfirmedAmount>
    <offlineUnconfirmedDonationCount>2</offlineUnconfirmedDonationCount>
    <onlineAmount>6500</onlineAmount>
    <onlineDonationCount>4</onlineDonationCount>
    <personalGoal>30000</personalGoal>
    <totalAmountIncludingUnconfirmed>7500</totalAmountIncludingUnconfirmed>
    <totalConfirmedAmount>6750</totalConfirmedAmount>
    <totalConfirmedDonationCount>5</totalConfirmedDonationCount>
    <totalDonationCountIncludingUnconfirmed>7</
totalDonationCountIncludingUnconfirmed>
  </fundraisingRecord>
</getFundraisingResponse>
```

JSON response

```
{"getFundraisingResponse":{"fundraisingRecord":
{"offlineConfirmedAmount":"250","personalGoal":"30000","offlineConfirmedDonationCount":"1
```

getGift

Provides gift detail information.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getGift &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&gift_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

gift_id

Required. Unique gift ID

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getGiftResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <gift>
    <acknowledged>false</acknowledged>
    <checkNumber xsi:nil="true"/>
    <confirmed>true</confirmed>
    <date>2013-02-03T05:18:26.996-06:00</date>
    <email>name1@example.com</email>
    <giftAmount>2501</giftAmount>
    <giftAidStatus>1</giftAidStatus>
    <giftMessage>Sample gift message.</giftMessage>
    <id>12001</id>
    <isRecurring>false</isRecurring>
    <name>
      <first>Jo</first>
      <middle>1</middle>
      <last>Public</last>
    </name>
    <paymentType>CREDIT</paymentType>
    <recognitionName>Sample recognition name 1</recognitionName>
  </gift>
</getGiftResponse>
```

JSON response

```
{"getGiftResponse":{"gift":{"giftMessage":"Sample gift
 message.","isRecurring":"false","paymentType":"CREDIT","checkNumber":null,"date":"2013-0
 recognition name
 1","confirmed":"true","email":"name1@example.com","giftAidStatus":"1","acknowledged":"fa
{"last":"Public","middle":"1","first":"Jo"},"giftAmount":"2501"}}}
```

getGiftCategories

Returns a list of the available gift categories associated to an event.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getGiftCategories`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getGiftCategoriesResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <giftCategory>
    <id>1</id>
    <name>sample name</name>
  </giftCategory>
</getGiftCategoriesResponse>
```

JSON response

```
{"getGiftCategoriesResponse":{"giftCategory":{"id":"1","name":"sample name"}}}
```

getGifts

Provides a list of gifts.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getGifts`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [

`&redirect=`*value* [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&cons_id=`*value* `&fr_id=`*value* [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Required. A unique constituent ID.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getGiftsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>2</totalNumberResults>
  <gift>
    <acknowledged>false</acknowledged>
    <checkNumber xsi:nil="true"/>
    <confirmed>true</confirmed>
    <date>2013-02-03T05:18:26.933-06:00</date>
    <email>name1@example.com</email>
    <giftAmount>2501</giftAmount>
    <giftAidStatus>1</giftAidStatus>
    <giftMessage>Sample gift message.</giftMessage>
    <id>12001</id>
    <isRecurring>false</isRecurring>
    <name>
      <first>Jo</first>
      <middle>1</middle>
      <last>Public</last>
    </name>
    <paymentType>CREDIT</paymentType>
    <recognitionName>Sample recognition name 1</recognitionName>
  </gift>
  <gift>
    <acknowledged>false</acknowledged>
    <checkNumber xsi:nil="true"/>
    <confirmed>true</confirmed>
    <date>2013-02-03T05:18:26.933-06:00</date>
    <email>name1@example.com</email>
    <giftAmount>2502</giftAmount>
    <giftAidStatus>1</giftAidStatus>
    <giftMessage>Sample gift message.</giftMessage>
    <id>12002</id>
    <isRecurring>false</isRecurring>
    <name>
      <first>Jo</first>
      <middle>2</middle>
      <last>Public</last>
    </name>
    <paymentType>CREDIT</paymentType>
```

```
    <recognitionName>Sample recognition name 2</recognitionName>
  </gift>
</getGiftsResponse>
```

JSON response

```
{"getGiftsResponse":{"gift":[{"giftMessage":"Sample gift
 message.","isRecurring":"false","paymentType":"CREDIT","checkNumber":null,"date":"2013-0
 recognition name
 1","confirmed":"true","email":"name1@example.com","giftAidStatus":"1","acknowledged":"fa
{"last":"Public","middle":"1","first":"Jo"},"giftAmount":"2501"},
{"giftMessage":"Sample gift
 message.","isRecurring":"false","paymentType":"CREDIT","checkNumber":null,"date":"2013-0
 recognition name
 2","confirmed":"true","email":"name1@example.com","giftAidStatus":"1","acknowledged":"fa
{"last":"Public","middle":"2","first":"Jo"},"giftAmount":"2502"}],"totalNumberResults":"2
```

getMessageLayouts

Provides a list of available message layouts or stationeries.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getMessageLayouts`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getMessageLayoutsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <layout>
    <layoutId>100</layoutId>
    <name>Sample layout 0</name>
    <thumbnailUrl>http://example.com/sample0.jpg</thumbnailUrl>
  </layout>
  <layout>
    <layoutId>101</layoutId>
    <name>Sample layout 1</name>
    <thumbnailUrl>http://example.com/sample1.jpg</thumbnailUrl>
  </layout>
  <layout>
    <layoutId>102</layoutId>
    <name>Sample layout 2</name>
    <thumbnailUrl>http://example.com/sample2.jpg</thumbnailUrl>
  </layout>
</getMessageLayoutsResponse>
```

JSON response

```
{"getMessageLayoutsResponse":{"layout":[{"thumbnailUrl":"http://
example.com/sample0.jpg","name":"Sample layout 0","layoutId":"100"},
{"thumbnailUrl":"http://example.com/sample1.jpg","name":"Sample
 layout 1","layoutId":"101"},{"thumbnailUrl":"http://example.com/
sample2.jpg","name":"Sample layout 2","layoutId":"102"}]}}
```

getNewsFeeds

Returns the requested number of PC2 admin newsfeed messages and total message count designated for the supplied fr_id and groups the requesting constituent or supplied cons_id is a member of.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getNewsFeeds`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&feed_count=`*value*] [`&feed_id=`*value*] [`&last_pc2_login=`*value*] [`&list_ascending=`*value*] [`&list_sort_column=`*MODIFY_DATE* | *CREATION_DATE* | *FEED_TITLE* | *STATUS* | *PUBLISHED_DATE* | *UNPUBLISHED_DATE* | *FEED_NAME*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

feed_count

Optional. The maximum number of newsfeed messages that will be returned.

Type xsd:integer.

Default is -1.

feed_id

Optional. The unique id of the message to be returned.

Type xsd:integer.

Default is -1.

last_pc2_login

Optional. The time of last pc2 login for the participant requesting news feeds.

Type xsd:nonNegativeInteger.

Default is 0.

list_ascending

Optional. If true, the returned list will be sorted in ascending order, if false, descending order. Default is false.

Type xsd:boolean.

Default is false.

list_sort_column

Optional. The attribute used to sort a list of newsfeed messages.

Type xsd:string.

Default is MODIFY_DATE.

Options are:

- MODIFY_DATE
- CREATION_DATE
- FEED_TITLE
- STATUS
- PUBLISHED_DATE
- UNPUBLISHED_DATE
- FEED_NAME

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getNewsFeedsResponse xmlns="http://convio.com/crm/v1.0" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
```

```
  <numberOfFeeds xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd">2</numberOfFeeds>
  <numberOfNewFeeds>1</numberOfNewFeeds>
  <newsFeed>
    <feedId>1000</feedId>
    <feedTitle>Example Newsfeed Message Title 0</feedTitle>
    <feedContent>Example Newsfeed Message Body 0</feedContent>
    <imageSrc>url/to/image0.jpg</imageSrc>
    <imageAltText>image alt text 0</imageAltText>
    <isNew>true</isNew>
    <lastModifiedDate>12/25/2010</lastModifiedDate>
    <mediaSrcs>
      <videoSource>http://www.youtube.com</videoSource>
      <videoDescription>youtube</videoDescription>
    </mediaSrcs>
    <feedType>Event</feedType>
    <isCaptainsOnly>false</isCaptainsOnly>
  </newsFeed>
  <newsFeed>
    <feedId>1001</feedId>
    <feedTitle>Example Newsfeed Message Title 1</feedTitle>
    <feedContent>Example Newsfeed Message Body 1</feedContent>
    <imageSrc>url/to/image1.jpg</imageSrc>
    <imageAltText>image alt text 1</imageAltText>
    <isNew>false</isNew>
    <lastModifiedDate>12/25/2010</lastModifiedDate>
    <mediaSrcs>
      <videoSource>http://www.youtube.com</videoSource>
      <videoDescription>youtube</videoDescription>
    </mediaSrcs>
    <feedType>Event</feedType>
    <isCaptainsOnly>false</isCaptainsOnly>
  </newsFeed>
</getNewsFeedsResponse>
```

JSON response

```
{"getNewsFeedsResponse":{"newsFeed":[{"feedId":"1000","feedTitle":"Example
 Newsfeed Message Title 0","imageSrc":"url/to/
image0.jpg","isCaptainsOnly":"false","feedType":"Event","isNew":"true","lastModifiedDate"
 Newsfeed Message Body 0","mediaSrcs":{"videoSource":"http://
www.youtube.com","videoDescription":"youtube"},"imageAltText":"image
 alt text 0"},{"feedId":"1001","feedTitle":"Example
 Newsfeed Message Title 1","imageSrc":"url/to/
image1.jpg","isCaptainsOnly":"false","feedType":"Event","isNew":"false","lastModifiedDate
 Newsfeed Message Body 1","mediaSrcs":{"videoSource":"http://
www.youtube.com","videoDescription":"youtube"},"imageAltText":"image alt
 text 1"}],"numberOfFeeds":{"content":"2","xsi:schemaLocation":"http://
convio.com/crm/v1.0 http://service.convio.net/xmlschema/
crm.public.v1.xsd"},"numberOfNewFeeds":"1"}}
```

getOrganizationMessage

Provides a rendered HTML block that that an organization wishes to display in the participant center.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getOrganizationMessage &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getOrganizationMessageResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <message>&lt;p>Sample HTML organization message.&lt;/p></message>
</getOrganizationMessageResponse>
```

JSON response

```
{"getOrganizationMessageResponse":{"message":"<p>Sample HTML organization
 message.<\/p>"}}
```

getParticipantCenterWrapper

Provides a rendered HTML block for use at the top of the participant center as well as related links.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getParticipantCenterWrapper &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*

Usage Notes

This method can be called anonymously, however, if you do not call this method within the context of a logged-in user, you must specify a valid Constituent ID as a cons_id parameter; otherwise the method will return an error.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getParticipantCenterWrapperResponse xsi:schemaLocation="http://convio.com/
crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <wrapper>
    <eventId>1234</eventId>
    <logoutUrl>http://example.com/logouturl</logoutUrl>
    <helpLinkUrl>http://example.com/helpurl</helpLinkUrl>
    <personalPageUrl>http://example.com/personalpageurl</personalPageUrl>
    <teamPageUrl>http://example.com/teampageurl</teamPageUrl>
    <companyPageUrl>http://example.com/companyurl</companyPageUrl>
    <organizationName>Sample Organization Name</organizationName>
    <eventName>Sample Event Name</eventName>
    <teamName>Sample Team Name</teamName>
    <companyName>Sample Company Name</companyName>
    <content>&lt;P>Sample HTML wrapper content.&lt;/P></content>
    <shareProvider>FACEBOOK</shareProvider>
    <shareProvider>TWITTER</shareProvider>
    <shareTitle>Support me</shareTitle>
    <shareAction>Donate</shareAction>
    <shareMessage>View my page</shareMessage>
    <shareId>EDP content</shareId>
  </wrapper>
</getParticipantCenterWrapperResponse>
```

JSON response

```
{"getParticipantCenterWrapperResponse":{"wrapper":{"teamName":"Sample Team
 Name","logoutUrl":"http://example.com/logouturl","organizationName":"Sample
 Organization Name","shareTitle":"Support me","companyName":"Sample Company
 Name","shareProvider":["FACEBOOK","TWITTER"],"content":"<P>Sample HTML
 wrapper content.<\/P>","shareMessage":"View my page","teamPageUrl":"http://
example.com/teampageurl","companyPageUrl":"http://example.com/
companyurl","shareId":"EDP content","eventId":"1234","helpLinkUrl":"http://
example.com/helpurl","personalPageUrl":"http://example.com/
personalpageurl","eventName":"Sample Event Name","shareAction":"Donate"}}}
```

getParticipantFBConnectInfo

Retrieves various info that is used by the Facebook FB Connect feature.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getParticipantFBConnectInfo &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*

][&sign_redirects=*value*][&source=*value*][&sub_source=*value*][&success_redirect=*value*] [&suppress_response_codes=*value*] &fr_id=*value*

Usage Notes

This method can be called anonymously, however, if you do not call this method within the context of a logged-in user, you must specify a valid Constituent ID as a cons_id parameter; otherwise the method will return an error.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getParticipantFBConnectInfoResponse xsi:schemaLocation="http://convio.com/
crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <teamName>teamname</teamName>
  <eventName>eventname</eventName>
  <personalPageUrl>http://www.convio.com</personalPageUrl>
  <teamPageUrl>http://www.convio.com</teamPageUrl>
</getParticipantFBConnectInfoResponse>
```

JSON response

```
{"getParticipantFBConnectInfoResponse":
{"teamName":"teamname","teamPageUrl":"http://
www.convio.com","personalPageUrl":"http://
www.convio.com","eventName":"eventname"}}
```

getParticipantProgress

Provides information about a participants personal and team fundraising progress.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

https://*secure2.convio.net/organization*/site/CRTeamraiserAPI?method=getParticipantProgress &api_key=*value* &v=*value* [&center_id=*value*][&error_redirect=*value*][&redirect=*value*][&response_format=*xml* | *json*][&sign_redirects=*value*][&source=*value*][&sub_source=*value*][&success_redirect=*value*] [&suppress_response_codes=*value*] &cons_id=*value* &fr_id=*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Required. A unique constituent ID.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getParticipantProgressResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <personalProgress>
    <goal>10000</goal>
    <raised>5600</raised>
    <percent>56</percent>
    <giftAidMatch>2000</giftAidMatch>
  </personalProgress>
  <teamProgress>
    <goal>25000</goal>
    <raised>12500</raised>
    <percent>50</percent>
  </teamProgress>
  <companyProgress>
    <goal>50000</goal>
    <raised>12500</raised>
    <percent>25</percent>
  </companyProgress>
  <daysLeft>7</daysLeft>
</getParticipantProgressResponse>
```

JSON response

```
{"getParticipantProgressResponse":{"teamProgress":
{"percent":"50","raised":"12500","goal":"25000"},"personalProgress":
{"percent":"56","raised":"5600","giftAidMatch":"2000","goal":"10000"},"daysLeft":"7","cor
{"percent":"25","raised":"12500","goal":"50000"}}}
```

getParticipants

Provides a list of participants in the specified TeamRaiser event or events of the specified Event Type that match the specified criteria. Search criteria includes the participant's first name, last name and/or team name.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getParticipants`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]

[`&event_type=`*value*][`&event_type2=`*value*][`&event_type3=`*value*][`&first_name=`*value*][`&fr_id=`*value*][`&full_search=`*value*][`&last_name=`*value*][`&list_ascending=`*value*][`&list_category_id=`*value*][`&list_filter_column=`*value*][`&list_filter_text=`*value*][`&list_page_offset=`*value*][`&list_page_size=`*value*][`&list_record_offset=`*value*][`&list_sort_column=`*value*][`&public_event_type=`*value*][`&screenname=`*value*][`&team_name=`*value*]

Usage Notes

This **client-only** method returns a list of active TeamRaiser Participants within a specified event that match the specified search criteria between First Name and/or Last Name. Unless full_search is TRUE, the specified TeamRaiser event must be active. An active participant is one who has not been marked as Inactive by an Administrator. An active event is an event that is accepting gifts, accepting registrations, or accepting gifts and registrations.

When searching on participant name and email, the search is a "like" search, and is case insensitive. A search for `first_name=jo` will return participants with a First name of "Jon", "John", "Joan", etc.

Note: This method may take up to 15 minutes following registration to return all pertinent data for a newly-registered participant.

Note: You must supply fr_id in order to retrieve performance data (amountRaised).

When using this method, you must provide a value for at least 1 of the following parameters: first_name or last_name. Each parameter is individually optional, but at least one of them must provide some data for the system to search on. If you do not provide a value to search on for one of those parameters, you will receive an error.

The search criteria must include a certain number of characters in order for the system to be able to perform the search. The default minimum characters comes from the Site Data Parameter `F2F_PART_FINDER_MIN_CRITERIA`. If that SDP is not set to a number, the search defaults to requiring a minimum of 3 characters. This minimum number of characters is applied to a combination of the first_name and last_name parameters.

Note: If the minimum characters is 3, then you must provide at least 3 characters total between the first_name and last_name parameters. A search for `first_name=n&last_name=ng` would be successful, but a search for `first_name=n&last_name=n` would generate an error.

Additionally, the list of Participants that is returned supports several parameters allowing finer control of the results, including filtering on any of the returned elements, pagination, and sorting.

Paginating the List takes two parameters: list_page_size, which specifies the number of results to be retrieved for that page. You must also provide **one** of the following two parameters:

- list_page_offset: specifies which page of the results to retrieve. The first page of results has a list_page_offset of 0, page 2 has a list_page_offset of 1, and so on.
- list_record_offset: specifies which record the page should start with. This also uses numbering from 0, so the first record has a list_record_offset of 0, the 2nd record has a list_record_offset of 1, and so on.

Filtering the List takes two parameters:

- list_filter_column: name of the element to be filtered on.
- list_filter_text: text that the element specified by list_filter_column must contain.

Note: list_filter_text performs an exact match, and is case sensitive.

Sorting the List takes two parameters:

- list_sort_column: name of the element to be sorted by.
- list_ascending: Boolean value indicating whether the sort should be in ascending order

Note: You can paginate, filter, and sort the list all in one call.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

event_type

Optional. The name of the Primary Event Type to limit search results to. Omit this parameter to search on all Primary Event Types. Defaults to null.

Type xsd:string.

event_type2

Optional. The name of the Event Type 2 to limit search results to. Omit this parameter to search on events with any Event Type 2. Defaults to null.

Type xsd:string.

event_type3

Optional. The name of the Event Type 3 to limit search results to. Omit this parameter to search on events with any Event Type 3. Defaults to null.

Type xsd:string.

first_name

Optional. Constituent first name.

Type xsd:string.

fr_id

Optional. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

full_search

Optional. If TRUE, returns all events regardless of event status, otherwise returns only published events. Possible status values include: Unpublished = 0, Accepting Registrations Only = 1, AcceptingRegistrations and Gifts = 2, Accepting Gifts Only= 3, Closed = 4, Archived = 8, Deleted = 9.

Type xsd:boolean.

last_name

Optional. Constituent last name.

Type xsd:string.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

public_event_type

Optional. The case-sensitive name of the Public Event Type to limit search results to. Omit this parameter to search on events with any Public Event Type. Defaults to null.

Type xsd:string.

screenname

Optional. Constituent screen name.

Type xsd:string.

team_name

Optional. A unique team name.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getParticipantsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>3</totalNumberResults>
  <participant>
    <name>
      <first>Participant</first>
      <last>Sample01</last>
```

```
    </name>
    <aTeamCaptain>false</aTeamCaptain>
    <teamName/>
    <donationUrl>http://example.com/donateurl</donationUrl>
    <personalPageUrl>http://example.com/personalpageurl</personalPageUrl>
    <teamPageUrl/>
    <consId>1001000</consId>
    <personalPagePrivate>false</personalPagePrivate>
    <amountRaised>5180</amountRaised>
    <goal>105060</goal>
  </participant>
  <participant>
    <name>
      <first>Participant</first>
      <last>Sample11</last>
    </name>
    <aTeamCaptain>false</aTeamCaptain>
    <teamName/>
    <donationUrl>http://example.com/donateurl</donationUrl>
    <personalPageUrl>http://example.com/personalpageurl</personalPageUrl>
    <teamPageUrl/>
    <consId>1001001</consId>
    <personalPagePrivate>false</personalPagePrivate>
    <amountRaised>5181</amountRaised>
    <goal>105061</goal>
  </participant>
  <participant>
    <name>
      <first>Participant</first>
      <last>Sample21</last>
    </name>
    <aTeamCaptain>true</aTeamCaptain>
    <teamName>Sample Team Name</teamName>
    <donationUrl>http://example.com/donateurl</donationUrl>
    <personalPageUrl>http://example.com/personalpageurl</personalPageUrl>
    <teamPageUrl>http://example.com/teampageurl</teamPageUrl>
    <consId>1001002</consId>
    <personalPagePrivate>false</personalPagePrivate>
    <amountRaised>5182</amountRaised>
    <goal>105062</goal>
  </participant>
</getParticipantsResponse>
```

JSON response

```
{"getParticipantsResponse":{"totalNumberResults":"3","participant":
[{"personalPagePrivate":"false","teamName":
{},"aTeamCaptain":"false","teamPageUrl":
{},"consId":"1001000","donationUrl":"http://example.com/donateurl","name":
{"last":"Sample01","first":"Participant"},"amountRaised":"5180","personalPageUrl":"http:/
example.com/personalpageurl","goal":"105060"},
{"personalPagePrivate":"false","teamName":
{},"aTeamCaptain":"false","teamPageUrl":
{},"consId":"1001001","donationUrl":"http://example.com/donateurl","name":
{"last":"Sample11","first":"Participant"},"amountRaised":"5181","personalPageUrl":"http:/
example.com/personalpageurl","goal":"105061"},
{"personalPagePrivate":"false","teamName":"Sample Team
 Name","aTeamCaptain":"true","teamPageUrl":"http://example.com/
teampageurl","consId":"1001002","donationUrl":"http://example.com/
donateurl","name":
```

```
{"last":"Sample21","first":"Participant"},"amountRaised":"5182","personalPageUrl":"http:/
example.com/personalpageurl","goal":"105062"}]}}
```

Examples

To perform an search for all participants named "Jane" in the TeamRaiser Event with the fr_id=1010:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getParticipants&first_name=jane&fr_id=1010
```

Searching the event with fr_id=1010, looking for participants with a first name containing "J", a last name containing "Doe", and only retrieving the first page of results, displaying 10 results per page:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
method=getParticipants&api_key=ACSAPI&response_format=xml&v=1.0&fr_id=1000&first_name=j&l
```

getParticipationType

Provides detailed information about a participation type.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getParticipationType &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&participation_type_id=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

participation_type_id

Optional. The unique ID of the participation type. If none is provided, the type you registered for will be used.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getParticipationTypeResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <participationType>
    <name>Sample Participation Type</name>
    <description>Sample Description</description>
    <fee>$10.00</fee>
    <id>1323</id>
```

```
  </participationType>
</getParticipationTypeResponse>
```

JSON response

```
{"getParticipationTypeResponse":{"participationType":
{"id":"1323","fee":"$10.00","description":"Sample Description","name":"Sample
 Participation Type"}}}
```

getParticipationTypes

Get the participation types defined for a TeamRaiser.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	SRTeamraiserAPI	GET, POST	Server.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getParticipationTypes &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*
Server API Syntax
`https://`*secure2.convio.net/organization*`/site/SRTeamraiserAPI?method=getParticipationTypes & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* |*json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& fr_id =`*value*

Usage Notes

`getParticipationTypes` retrieves a list of the participation types associated with a TeamRaiser and includes detailed information about each participation type.

`getParticipationTypes` can be used in conjunction with *getRegistrationDocument* and *processRegistration* to register a participant in a TeamRaiser.

Additional documentation of each element of the response is available in the *TeamRaiser XML Schema Definition*.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getParticipationTypesResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <participationType>
    <name>Shark Battler</name>
    <description>A shark battler will support shark awareness by battling two
 great whites in hand-to-hand combat.</description>
    <fee>$10.00</fee>
    <rawFee>1000</rawFee>
    <id>1000</id>
    <securityCategory>0</securityCategory>
    <order>0</order>
    <active>true</active>
    <standardRegistrationAllowed>true</standardRegistrationAllowed>
    <anonymousRegistrationAllowed>true</anonymousRegistrationAllowed>
    <screenNameRegistrationAllowed>true</screenNameRegistrationAllowed>
    <taxDeductibleAmountOfFee>$5.00</taxDeductibleAmountOfFee>
    <rawTaxDeductibleAmountOfFee>500</rawTaxDeductibleAmountOfFee>
    <minimumGoal>$10.00</minimumGoal>
    <rawMinimumGoal>1000</rawMinimumGoal>
    <promoCodeRequired>false</promoCodeRequired>
    <waiver required="false">I understand that it is an incredibly bad idea
 for me to do this. I don't care.</waiver>
    <surveyQuestion>
      <questionText>Can you bring your own harpoon and wetsuit?</questionText>
      <questionId>1100</questionId>
      <questionRequired>true</questionRequired>
      <questionOrderNumber>1</questionOrderNumber>
      <questionMinResponses xsi:nil="true"/>
      <questionMaxResponses xsi:nil="true"/>
      <key>t_shirt</key>
      <categoryId>-1</categoryId>
      <surveyInstanceId>1200</surveyInstanceId>
      <questionType>YesNo</questionType>
      <hidden>false</hidden>
    </surveyQuestion>
    <captainQuestion>
      <questionText>Are you prepared to lead your members into glory?</
questionText>
      <questionId>1101</questionId>
      <questionRequired>true</questionRequired>
      <questionOrderNumber>2</questionOrderNumber>
      <questionMinResponses xsi:nil="true"/>
      <questionMaxResponses xsi:nil="true"/>
      <key>glory</key>
      <surveyInstanceId>1200</surveyInstanceId>
      <questionType>YesNo</questionType>
      <hidden>false</hidden>
      <categoryId>-1</categoryId>
    </captainQuestion>
    <defaultMailOptIn>true</defaultMailOptIn>
    <defaultEmailOptIn>true</defaultEmailOptIn>
    <primaryRegistrationFields>
      <field required="true">firstName</field>
      <field required="true">lastName</field>
```

```
      <field required="true">email</field>
      <field required="true">firstName</field>
      <field required="true">lastName</field>
      <field required="true">email</field>
      <field>emergencyContact</field>
      <field>phoneNumber</field>
      <field>title</field>
      <field>middleName</field>
      <field>suffix</field>
      <field>professionalSuffix</field>
      <field>address</field>
      <field>phoneNumber</field>
      <field>dateOfBirth</field>
      <field>gender</field>
      <field>employerName</field>
      <field>occupation</field>
      <field>honoree</field>
   </primaryRegistrationFields>
   <secondaryRegistrationFields>
      <field required="true">firstName</field>
      <field required="true">lastName</field>
      <field>email</field>
      <field required="true">firstName</field>
      <field required="true">lastName</field>
      <field required="true">email</field>
      <field>emergencyContact</field>
      <field>phoneNumber</field>
      <field>title</field>
      <field>middleName</field>
      <field>suffix</field>
      <field>professionalSuffix</field>
      <field>address</field>
      <field>phoneNumber</field>
      <field>dateOfBirth</field>
      <field>gender</field>
      <field>employerName</field>
      <field>occupation</field>
      <field>honoree</field>
   </secondaryRegistrationFields>
   <discountCodeAvailable>false</discountCodeAvailable>
   <emailOptInText>Yes, I would like to receive email from American Health
Society</emailOptInText>
   <postalMailOptInText>Yes, I would like to receive postal mail from
American Health Society</postalMailOptInText>
 </participationType>
 <participationType>
   <name>Shark Defender</name>
   <description>A shark defender will support shark awesomeness by harpooning
shark battlers, riding sharks around and assisting in sealion hunts and
BBQs.</description>
   <fee>$0.00</fee>
   <rawFee>0</rawFee>
   <id>1001</id>
   <securityCategory>1</securityCategory>
   <order>1</order>
   <active>true</active>
   <standardRegistrationAllowed>true</standardRegistrationAllowed>
   <anonymousRegistrationAllowed>true</anonymousRegistrationAllowed>
   <screenNameRegistrationAllowed>true</screenNameRegistrationAllowed>
   <taxDeductibleAmountOfFee>$5.00</taxDeductibleAmountOfFee>
   <rawTaxDeductibleAmountOfFee>500</rawTaxDeductibleAmountOfFee>
   <minimumGoal>$100.00</minimumGoal>
   <rawMinimumGoal>10000</rawMinimumGoal>
   <promoCodeRequired>false</promoCodeRequired>
```

```
    <waiver required="true">I vow complete annhiliation of shark hunters and
 occasionally bringing a 6-pack to the BBQ.</waiver>
    <upsellId>1200</upsellId>
    <upsellId>1201</upsellId>
    <surveyQuestion>
      <questionText>Can you bring your own harpoon and wetsuit?</questionText>
      <questionId>1100</questionId>
      <questionRequired>true</questionRequired>
      <questionOrderNumber>1</questionOrderNumber>
      <questionMinResponses xsi:nil="true"/>
      <questionMaxResponses xsi:nil="true"/>
      <key>t_shirt</key>
      <categoryId>1</categoryId>
      <surveyInstanceId>1200</surveyInstanceId>
      <questionType>YesNo</questionType>
      <hidden>true</hidden>
    </surveyQuestion>
    <defaultMailOptIn>true</defaultMailOptIn>
    <defaultEmailOptIn>true</defaultEmailOptIn>
    <participationTypeRegistrationLimit>
      <registrationLimit>1000</registrationLimit>
      <currentRegistrations>990</currentRegistrations>
      <limitReached>true</limitReached>
      <makeNewSlotsUnavailable>true</makeNewSlotsUnavailable>
      <hideNumberLeft>false</hideNumberLeft>
    </participationTypeRegistrationLimit>
  </participationType>
</getParticipationTypesResponse>
```


JSON response


```
{"getParticipationTypesResponse":{"participationType":
[{"rawTaxDeductibleAmountOfFee":"500","surveyQuestion":
{"questionMaxResponses":null,"questionMinResponses":null,"questionRequired":"true","quest
 you bring your own harpoon and
 wetsuit?","questionOrderNumber":"1","questionType":"YesNo","surveyInstanceId":"1200","ke
{"content":"I understand that it is an incredibly bad idea for me to do this.
 I don't
 care.","required":"false"},"standardRegistrationAllowed":"true","secondaryRegistrationFi
{"field":[{"content":"firstName","required":"true"},
{"content":"lastName","required":"true"},"email",
{"content":"firstName","required":"true"},
{"content":"lastName","required":"true"},
{"content":"email","required":"true"},"emergencyContact","phoneNumber","title","middleNam
 shark battler will support shark awareness by battling
 two great whites in hand-to-hand combat.","name":"Shark
 Battler","minimumGoal":"$10.00","captainQuestion":
{"questionMaxResponses":null,"questionMinResponses":null,"questionRequired":"true","quest
 you prepared to lead your members into
 glory?","questionOrderNumber":"2","questionType":"YesNo","surveyInstanceId":"1200","key"
{"field":[{"content":"firstName","required":"true"},
{"content":"lastName","required":"true"},
{"content":"email","required":"true"},
{"content":"firstName","required":"true"},
{"content":"lastName","required":"true"},
{"content":"email","required":"true"},"emergencyContact","phoneNumber","title","middleNam
 I would like to receive postal mail from American Health
 Society","rawMinimumGoal":"1000","fee":"$10.00","anonymousRegistrationAllowed":"true","c
 I would like to receive email from American Health Society"},
{"rawTaxDeductibleAmountOfFee":"500","defaultEmailOptIn":"true","surveyQuestion":
{"questionMaxResponses":null,"questionMinResponses":null,"questionRequired":"true","quest
```

```
 you bring your own harpoon and
 wetsuit?","questionOrderNumber":"1","questionType":"YesNo","surveyInstanceId":"1200","ke
{"registrationLimit":"1000","hideNumberLeft":"false","limitReached":"true","makeNewSlotsU
{"content":"I vow complete annhiliation of shark
 hunters and occasionally bringing a 6-pack to the
 BBQ.","required":"true"},"standardRegistrationAllowed":"true","rawFee":"0","upsellId":
["1200","1201"],"rawMinimumGoal":"10000","securityCategory":"1","fee":"$0.00","id":"1001"
 shark defender will support shark awesomeness by harpooning
 shark battlers, riding sharks around and assisting in sealion
 hunts and BBQs.","taxDeductibleAmountOfFee":"$5.00","name":"Shark
 Defender","active":"true","minimumGoal":"$100.00","promoCodeRequired":"false"}]}}
```

getPersonalDonationByDay

Provides data used to build daily progress charts.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getPersonalDonationByDay &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getPersonalDonationByDayResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <donationDataByDay>
    <date>2013-02-02-06:00</date>
    <amount>3000</amount>
  </donationDataByDay>
```

```
  <donationDataByDay>
    <date>2013-02-03-06:00</date>
  </donationDataByDay>
</getPersonalDonationByDayResponse>
```

JSON response

```
{"getPersonalDonationByDayResponse":{"donationDataByDay":
[{"amount":"3000","date":"2013-02-02-06:00"},{"date":"2013-02-03-06:00"}]}}
```

getPersonalMediaLayout

Retrieves a participants personal media layout

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getPersonalMediaLayout &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getPersonalMediaLayoutResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <personalMediaLayout>photos</personalMediaLayout>
</getPersonalMediaLayoutResponse>
```

JSON response

```
{"getPersonalMediaLayoutResponse":{"personalMediaLayout":"photos"}}
```

getPersonalPageInfo

Provides information used to display or edit a personal page.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getPersonalPageInfo &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&cons_id=`*value* `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Required. A unique constituent ID.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getPersonalPageResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <personalPage>
    <pageTitle>Sample Personal Page Title</pageTitle>
    <richText>&lt;P>Sample personal page rich text.&lt;/P></richText>
    <richTextEnabled>true</richTextEnabled>
    <availableLayouts>
      <layout>Single Photo/Status left edge</layout>
      <layout>Single Photo/Status block inset</layout>
```

```
      <layout>Two Photos Left</layout>
      <layout>Two Photos Right</layout>
      <layout>Customizable</layout>
    </availableLayouts>
  </personalPage>
</getPersonalPageResponse>
```

JSON response

```
{"getPersonalPageResponse":{"personalPage":{"availableLayouts":{"layout":
["Single Photo/Status left edge","Single Photo/Status block inset","Two
 Photos Left","Two Photos Right","Customizable"]},"pageTitle":"Sample
 Personal Page Title","richText":"<P>Sample personal page rich text.<\/
P>","richTextEnabled":"true"}}}
```

getPersonalPagePreview

Provides a preview of the personal page.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getPersonalPagePreview &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&blog_enabled=`*value*][`&headline1=`*value*][`&headline2=`*value*][`&headline3=`*value*][`&page_layout=`*value*][`&page_title=`*value*][`&rich_text=`*value*][`&status1=`*value*][`&status2=`*value*][`&text1=`*value*][`&text2=`*value*][`&text3=`*value*][`&auth=`*value*][`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

blog_enabled

Optional. Whether or not your personal page blog is enabled.

Type xsd:boolean.

headline1

Optional. The first headline on the page.

Type xsd:string.

headline2

Optional. The second headline on the page.

Type xsd:string.

headline3

Optional. The third headline on the page.

Type xsd:string.

page_layout

Optional. The selected page layout

page_title

Optional. The title of the page

rich_text

Optional. The rich text area of the page.

Type xsd:string.

status1

Optional. Status Indicator 1

Type xsd:string.

status2

Optional. Status Indicator 2

Type xsd:string.

text1

Optional. The first text area on the page.

Type xsd:string.

text2

Optional. The second text area on the page.

Type xsd:string.

text3

Optional. The third text area on the page.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getPersonalPagePreviewResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <previewUrl>http://www.example.com/preview.html</previewUrl>
</getPersonalPagePreviewResponse>
```

JSON response

```
{"getPersonalPagePreviewResponse":{"previewUrl":"http://www.example.com/
preview.html"}}
```

getPersonalPhotos

Retrieves a participant's uploaded photos.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getPersonalPhotos &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getPersonalPhotosResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <photoItem>
    <id>1</id>
    <thumbnailUrl>&lt;url></thumbnailUrl>
    <customUrl>&lt;url></customUrl>
    <originalUrl>&lt;url></originalUrl>
  </photoItem>
</getPersonalPhotosResponse>
```

JSON response

```
{"getPersonalPhotosResponse":{"photoItem":
{"id":"1","originalUrl":"<url>","thumbnailUrl":"<url>","customUrl":"<url>"}}}
```

getPersonalVideoUrl

Retrieves a participant's video url.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getPersonalVideoUrl &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getPersonalVideoUrlResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <videoUrl>http://www.youtube.com/watch?v=3eZTh94Fapg</videoUrl>
</getPersonalVideoUrlResponse>
```

JSON response

```
{"getPersonalVideoUrlResponse":{"videoUrl":"http://www.youtube.com/watch?
v=3eZTh94Fapg"}}
```

getPublicEventTypes

Returns the details of all well-known TeamRaiser public event types that match the search criteria.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getPublicEventTypes &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&active_only=`*value*] [`&event_type=`*value*]

Usage Notes

This method was designed primarily to support the dynamic creation of TeamRaiser search forms via REST APIs. You could, for example, invoke this method to discover valid public event type names to be included in an HTTP search form.

Of particular importance when using this method is the relationship between **public** event types and **primary** event types. Every primary event type is associated with zero or more public event types. Conversely, a public event type is associated with exactly one primary event type. That relationship has important consequences for TeamRaiser search forms. TeamRaiser search forms that constrain search results by primary event type should pass that same primary event constraint to this method when requesting a list of valid public event types.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

active_only

Optional. A value of true indicates that only currently active public event types should be returned in the search results. Defaults to true.

Type xsd:boolean.

event_type

Optional. The name of the primary event type to limit search results to. Omit this parameter to search for public event types associated with any primary event type. Defaults to null.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getPublicEventTypesResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>3</totalNumberResults>
  <publicEventType>
    <publicEventTypeName>Marathon</publicEventTypeName>
    <primaryEventTypeName>Endurance Event</primaryEventTypeName>
    <active>true</active>
  </publicEventType>
  <publicEventType>
    <publicEventTypeName>Run/Walk</publicEventTypeName>
    <primaryEventTypeName>Endurance Event</primaryEventTypeName>
    <active>false</active>
  </publicEventType>
```

```
  <publicEventType>
    <publicEventTypeName>10 K</publicEventTypeName>
    <primaryEventTypeName>Endurance Event</primaryEventTypeName>
    <active>true</active>
  </publicEventType>
</getPublicEventTypesResponse>
```

JSON response

```
{"getPublicEventTypesResponse":{"publicEventType":
[{"publicEventTypeName":"Marathon","primaryEventTypeName":"Endurance
 Event","active":"true"},{"publicEventTypeName":"Run/
Walk","primaryEventTypeName":"Endurance Event","active":"false"},
{"publicEventTypeName":"10 K","primaryEventTypeName":"Endurance
 Event","active":"true"}],"totalNumberResults":"3"}}
```

Examples

To find a list of all valid, active public event types:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getPublicEventTypes
```

To request a list of active public event types that are appropriate for the 'Foo' primary event type:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getPublicEventTypes&event_type=Foo
```

To request a complete listing of active and non-active public event types:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getPublicEventTypes&active_only=false
```

getRecentActivity

Provides a list of recent activity for display in the participant center.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getRecentActivity`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getRecentActivityResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <activityRecord>
    <type>DONATION</type>
    <activity>Sample activity 01</activity>
    <followup>Sample followup 01</followup>
    <date>January 1</date>
    <contactId>100</contactId>
  </activityRecord>
  <activityRecord>
    <type>MESSAGE</type>
    <activity>Sample activity 11</activity>
    <followup>Sample followup 11</followup>
    <date>January 1</date>
    <contactId>101</contactId>
  </activityRecord>
  <activityRecord>
    <type>RECRUIT</type>
    <activity>Sample activity 21</activity>
    <followup>Sample followup 21</followup>
    <date>January 1</date>
    <contactId>102</contactId>
  </activityRecord>
  <activityRecord>
    <type>DONATION</type>
    <activity>Sample activity 31</activity>
    <followup>Sample followup 31</followup>
    <date>January 1</date>
    <contactId>103</contactId>
  </activityRecord>
  <activityRecord>
    <type>MESSAGE</type>
    <activity>Sample activity 41</activity>
    <followup>Sample followup 41</followup>
    <date>January 1</date>
    <contactId>104</contactId>
  </activityRecord>
  <activityRecord>
    <type>RECRUIT</type>
    <activity>Sample activity 51</activity>
    <followup>Sample followup 51</followup>
    <date>January 1</date>
    <contactId>105</contactId>
  </activityRecord>
  <activityRecord>
    <type>DONATION</type>
    <activity>Sample activity 61</activity>
    <followup>Sample followup 61</followup>
    <date>January 1</date>
    <contactId>106</contactId>
  </activityRecord>
```

```
  <activityRecord>
    <type>MESSAGE</type>
    <activity>Sample activity 71</activity>
    <followup>Sample followup 71</followup>
    <date>January 1</date>
    <contactId>107</contactId>
  </activityRecord>
  <activityRecord>
    <type>RECRUIT</type>
    <activity>Sample activity 81</activity>
    <followup>Sample followup 81</followup>
    <date>January 1</date>
    <contactId>108</contactId>
  </activityRecord>
  <activityRecord>
    <type>DONATION</type>
    <activity>Sample activity 91</activity>
    <followup>Sample followup 91</followup>
    <date>January 1</date>
    <contactId>109</contactId>
  </activityRecord>
</getRecentActivityResponse>
```

JSON response


```
{"getRecentActivityResponse":{"activityRecord":
[{"contactId":"100","followup":"Sample followup 01","date":"January
 1","type":"DONATION","activity":"Sample activity
 01"},{"contactId":"101","followup":"Sample followup
 11","date":"January 1","type":"MESSAGE","activity":"Sample
 activity 11"},{"contactId":"102","followup":"Sample followup
 21","date":"January 1","type":"RECRUIT","activity":"Sample
 activity 21"},{"contactId":"103","followup":"Sample followup
 31","date":"January 1","type":"DONATION","activity":"Sample
 activity 31"},{"contactId":"104","followup":"Sample followup
 41","date":"January 1","type":"MESSAGE","activity":"Sample
 activity 41"},{"contactId":"105","followup":"Sample followup
 51","date":"January 1","type":"RECRUIT","activity":"Sample
 activity 51"},{"contactId":"106","followup":"Sample followup
 61","date":"January 1","type":"DONATION","activity":"Sample
 activity 61"},{"contactId":"107","followup":"Sample followup
 71","date":"January 1","type":"MESSAGE","activity":"Sample
 activity 71"},{"contactId":"108","followup":"Sample followup
 81","date":"January 1","type":"RECRUIT","activity":"Sample activity
 81"},{"contactId":"109","followup":"Sample followup 91","date":"January
 1","type":"DONATION","activity":"Sample activity 91"}]}}
```


getRegisteredTeamraisers

Provides a list of events that a participant is registered for.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getRegisteredTeamraisers &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*

][&sign_redirects=*value*][&source=*value*][&sub_source=*value*][&success_redirect=*value*][&suppress_response_codes=*value*][&cons_id=*value*][&event_type=*value*][&event_type2=*value*][&event_type3=*value*][&public_event_type=*value*][&user_name=*value*]

Usage Notes

This method can be called anonymously, however, if you do not call this method within the context of a logged-in user, you must specify either a valid Constituent ID as a cons_id parameter or a valid user name as a user_name parameter; otherwise the method will return an error.

If the constituent does not belong to a team, i.e. they are registered as an individual, the teamName and teamPageUrl elements will be nil. If the constituent is an event manager, the eventManageUrl element will contain their event management center url; otherwise the eventManageUrl element will be nil.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Optional. This is the constituent ID of the user you want to see registered events for.

Type xsd:nonNegativeInteger.

event_type

Optional. The name of the Primary Event Type to limit search results to. Omit this parameter to search on all Primary Event Types. Defaults to null.

Type xsd:string.

event_type2

Optional. The name of the Event Type 2 to limit search results to. Omit this parameter to search on events with any Event Type 2. Defaults to null.

Type xsd:string.

event_type3

Optional. The name of the Event Type 3 to limit search results to. Omit this parameter to search on events with any Event Type 3. Defaults to null.

Type xsd:string.

public_event_type

Optional. The case-sensitive name of the Public Event Type to limit search results to. Omit this parameter to search on events with any Public Event Type. Defaults to null.

Type xsd:string.

user_name

Optional. This is the username of the user you want to see registered events for.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getRegisteredTeamraisersResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <consId>1234567</consId>
  <teamraiser>
```

```
    <id>12001</id>
    <name>Sample Event Title 1</name>
    <description>Sample Event Description 1</description>
    <event_date>2013-02-03T05:18:26.907-06:00</event_date>
    <accepting_registrations>true</accepting_registrations>
    <accepting_donations>true</accepting_donations>
    <location_name>Sample Location 1</location_name>
    <street_address>1234 Fake St.</street_address>
    <city>Austin</city>
    <state>TX</state>
    <zip>78758</zip>
    <county>Travis</county>
    <country>United States</country>
    <region>Hill Country</region>
    <area>North Austin</area>
    <mailingAddrAttn>J. Doe</mailingAddrAttn>
    <mail_street1>1234 Fake Ave.</mail_street1>
    <mail_street2>Suite 1000</mail_street2>
    <mail_city>Austin</mail_city>
    <mail_state>TX</mail_state>
    <mail_zip>78758</mail_zip>
    <status>1</status>
    <greeting_url>http://example.com/greetingUrl</greeting_url>
    <find_url>http://example.com/findTeamOrParticipantUrl</find_url>
    <donate_event_url>http://example.com/donateUrl</donate_event_url>
    <reg_indiv_url>http://example.com/registerIndividualUrl</reg_indiv_url>
    <reg_new_team_url>http://example.com/formANewTeamUrl</reg_new_team_url>
    <reg_join_team_url>http://example.com/joinTeamUrl</reg_join_team_url>
    <public_event_type_name>Central Texas</public_event_type_name>
    <ecommerce_event_url>http://example.com/eCommerceEventUrl</
ecommerce_event_url>
    <teamName>Sample Team Name 1</teamName>
    <teamPageUrl>http://www.sampleteam.com/page1</teamPageUrl>
    <eventManageUrl>http://www.samplecenter.com/center1</eventManageUrl>
  </teamraiser>
  <teamraiser>
    <id>12002</id>
    <name>Sample Event Title 2</name>
    <description>Sample Event Description 2</description>
    <event_date>2013-02-03T05:18:26.908-06:00</event_date>
    <accepting_registrations>true</accepting_registrations>
    <accepting_donations>true</accepting_donations>
    <location_name>Sample Location 2</location_name>
    <street_address>1234 Fake St.</street_address>
    <city>Austin</city>
    <state>TX</state>
    <zip>78758</zip>
    <county>Travis</county>
    <country>United States</country>
    <region>Hill Country</region>
    <area>North Austin</area>
    <mailingAddrAttn>J. Doe</mailingAddrAttn>
    <mail_street1>1234 Fake Ave.</mail_street1>
    <mail_street2>Suite 1000</mail_street2>
    <mail_city>Austin</mail_city>
    <mail_state>TX</mail_state>
    <mail_zip>78758</mail_zip>
    <status>1</status>
    <greeting_url>http://example.com/greetingUrl</greeting_url>
    <find_url>http://example.com/findTeamOrParticipantUrl</find_url>
    <donate_event_url>http://example.com/donateUrl</donate_event_url>
    <reg_indiv_url>http://example.com/registerIndividualUrl</reg_indiv_url>
    <reg_new_team_url>http://example.com/formANewTeamUrl</reg_new_team_url>
    <reg_join_team_url>http://example.com/joinTeamUrl</reg_join_team_url>
```

```
    <public_event_type_name>Central Texas</public_event_type_name>
    <ecommerce_event_url>http://example.com/eCommerceEventUrl</
ecommerce_event_url>
    <teamName>Sample Team Name 2</teamName>
    <teamPageUrl>http://www.sampleteam.com/page2</teamPageUrl>
    <eventManageUrl xsi:nil="true"/>
  </teamraiser>
  <teamraiser>
    <id>12003</id>
    <name>Sample Event Title 3</name>
    <description>Sample Event Description 3</description>
    <event_date>2013-02-03T05:18:26.908-06:00</event_date>
    <accepting_registrations>true</accepting_registrations>
    <accepting_donations>true</accepting_donations>
    <location_name>Sample Location 3</location_name>
    <street_address>1234 Fake St.</street_address>
    <city>Austin</city>
    <state>TX</state>
    <zip>78758</zip>
    <county>Travis</county>
    <country>United States</country>
    <region>Hill Country</region>
    <area>North Austin</area>
    <mailingAddrAttn>J. Doe</mailingAddrAttn>
    <mail_street1>1234 Fake Ave.</mail_street1>
    <mail_street2>Suite 1000</mail_street2>
    <mail_city>Austin</mail_city>
    <mail_state>TX</mail_state>
    <mail_zip>78758</mail_zip>
    <status>1</status>
    <greeting_url>http://example.com/greetingUrl</greeting_url>
    <find_url>http://example.com/findTeamOrParticipantUrl</find_url>
    <donate_event_url>http://example.com/donateUrl</donate_event_url>
    <reg_indiv_url>http://example.com/registerIndividualUrl</reg_indiv_url>
    <reg_new_team_url>http://example.com/formANewTeamUrl</reg_new_team_url>
    <reg_join_team_url>http://example.com/joinTeamUrl</reg_join_team_url>
    <public_event_type_name>Central Texas</public_event_type_name>
    <ecommerce_event_url>http://example.com/eCommerceEventUrl</
ecommerce_event_url>
    <teamName>Sample Team Name 3</teamName>
    <teamPageUrl>http://www.sampleteam.com/page3</teamPageUrl>
    <eventManageUrl>http://www.samplecenter.com/center3</eventManageUrl>
  </teamraiser>
</getRegisteredTeamraisersResponse>
```


JSON response


```
{"getRegisteredTeamraisersResponse":{"consId":"1234567","teamraiser":
[{"teamName":"Sample Team Name 1","region":"Hill
 Country","state":"TX","city":"Austin","id":"12001","reg_new_team_url":"http://
example.com/formANewTeamUrl","mail_state":"TX","area":"North
 Austin","eventManageUrl":"http://www.samplecenter.com/
center1","description":"Sample Event Description 1","name":"Sample
 Event Title 1","mail_city":"Austin","reg_indiv_url":"http://
example.com/registerIndividualUrl","ecommerce_event_url":"http://
example.com/eCommerceEventUrl","public_event_type_name":"Central
 Texas","zip":"78758","reg_join_team_url":"http://
example.com/joinTeamUrl","location_name":"Sample
 Location 1","status":"1","find_url":"http://example.com/
findTeamOrParticipantUrl","country":"United
 States","accepting_donations":"true","greeting_url":"http://
```

```
example.com/greetingUrl","donate_event_url":"http://
example.com/donateUrl","teamPageUrl":"http://www.sampleteam.com/
page1","county":"Travis","event_date":"2013-02-03T05:18:26.910-06:00","mail_zip":"78758",
 Doe","street_address":"1234 Fake
 St.","accepting_registrations":"true","mail_street1":"1234 Fake
 Ave.","mail_street2":"Suite 1000"},{"teamName":"Sample Team Name
 2","region":"Hill
 Country","state":"TX","city":"Austin","id":"12002","reg_new_team_url":"http://
example.com/formANewTeamUrl","mail_state":"TX","area":"North
 Austin","eventManageUrl":null,"description":"Sample
 Event Description 2","name":"Sample Event Title
 2","mail_city":"Austin","reg_indiv_url":"http://example.com/
registerIndividualUrl","ecommerce_event_url":"http://
example.com/eCommerceEventUrl","public_event_type_name":"Central
 Texas","zip":"78758","reg_join_team_url":"http://
example.com/joinTeamUrl","location_name":"Sample
 Location 2","status":"1","find_url":"http://example.com/
findTeamOrParticipantUrl","country":"United
 States","accepting_donations":"true","greeting_url":"http://
example.com/greetingUrl","donate_event_url":"http://
example.com/donateUrl","teamPageUrl":"http://www.sampleteam.com/
page2","county":"Travis","event_date":"2013-02-03T05:18:26.910-06:00","mail_zip":"78758",
 Doe","street_address":"1234 Fake
 St.","accepting_registrations":"true","mail_street1":"1234 Fake
 Ave.","mail_street2":"Suite 1000"},{"teamName":"Sample Team Name
 3","region":"Hill
 Country","state":"TX","city":"Austin","id":"12003","reg_new_team_url":"http://
example.com/formANewTeamUrl","mail_state":"TX","area":"North
 Austin","eventManageUrl":"http://www.samplecenter.com/
center3","description":"Sample Event Description 3","name":"Sample
 Event Title 3","mail_city":"Austin","reg_indiv_url":"http://
example.com/registerIndividualUrl","ecommerce_event_url":"http://
example.com/eCommerceEventUrl","public_event_type_name":"Central
 Texas","zip":"78758","reg_join_team_url":"http://
example.com/joinTeamUrl","location_name":"Sample
 Location 3","status":"1","find_url":"http://example.com/
findTeamOrParticipantUrl","country":"United
 States","accepting_donations":"true","greeting_url":"http://
example.com/greetingUrl","donate_event_url":"http://
example.com/donateUrl","teamPageUrl":"http://www.sampleteam.com/
page3","county":"Travis","event_date":"2013-02-03T05:18:26.911-06:00","mail_zip":"78758",
 Doe","street_address":"1234 Fake
 St.","accepting_registrations":"true","mail_street1":"1234 Fake
 Ave.","mail_street2":"Suite 1000"}]}}
```


getRegistration

Provides detailed registration information for a participant.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getRegistration`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getRegistrationResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <registration>
    <addonsTotal>2400</addonsTotal>
    <consId>1234567</consId>
    <date>2013-02-03T05:18:25.506-06:00</date>
    <emergencyName>Sample Emergency Name 1</emergencyName>
    <emergencyPhone>512-555-1234</emergencyPhone>
    <eventId>1001</eventId>
    <extraGift>1500</extraGift>
    <feePaid>1000</feePaid>
    <goal>15000</goal>
    <parentRegistrationId>2345</parentRegistrationId>
    <participationTypeId>3456</participationTypeId>
    <pledgeId>222333444</pledgeId>
    <privatePage>false</privatePage>
    <raceNumber>234</raceNumber>
    <registrationId>222222222</registrationId>
    <shortcutId>3333</shortcutId>
    <aTeamCaptain>false</aTeamCaptain>
    <teamId>4444</teamId>
    <tentId>5555</tentId>
    <tentmateConsId>1222222</tentmateConsId>
    <tentmateStatus>5</tentmateStatus>
    <transactionId>55555555</transactionId>
    <tentingAllowed>false</tentingAllowed>
    <flowStep>4</flowStep>
    <checkinStatus>committed</checkinStatus>
    <selfDonor>true</selfDonor>
    <teamInformation>
      <divisionName>Sample Division Name</divisionName>
      <password>Sample Password</password>
      <recruitingGoal>10</recruitingGoal>
    </teamInformation>
    <companyInformation>
      <companyId>0</companyId>
      <companyType>LOCAL</companyType>
    </companyInformation>
```

```
    <receiveGiftNotification>true</receiveGiftNotification>
    <previousEventParticipant>false</previousEventParticipant>
    <lastPC2Login>0</lastPC2Login>
  </registration>
</getRegistrationResponse>
```

JSON response

```
{"getRegistrationResponse":{"registration":
{"tentmateConsId":"1222222","transactionId":"55555555","flowStep":"4","pledgeId":"222333
{"companyType":"LOCAL","companyId":"0"},"receiveGiftNotification":"true","participationTy
 Emergency Name
 1","addonsTotal":"2400","aTeamCaptain":"false","tentmateStatus":"5","extraGift":"1500","
{"divisionName":"Sample Division Name","password":"Sample
 Password","recruitingGoal":"10"},"lastPC2Login":"0"}}}
```

getRegistrationDocument

Get template document for registering for a TeamRaiser.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	SRTeamraiserAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getRegistrationDocument &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&participation_id=`*value* [`&include_username_element=`*value*] [`&secondary_participation_id=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRTeamraiserAPI?method=getRegistrationDocument & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& fr_id =`*value* `& participation_id =`*value* [`& include_username_element =`*value*] [`& secondary_participation_id =`*value*]

Usage Notes

`getRegistrationDocument` retrieves an XML document that can be used as a template to register a participant of a TeamRaiser with the *processRegistration* method.

The registration document varies depending on the TeamRaiser and participation type. A list of participation types for a TeamRaiser can be retrieved with the *getParticipationTypes* method.

Additional documentation of each element of the response is available in the *TeamRaiser XML Schema Definition*.

Note: If the selected event and participation type combination is not currently accepting registrations, this method will return an error.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

participation_id

Required. The TeamRaiser participation type ID.

Type xsd:nonNegativeInteger.

include_username_element

Optional. Determines whether or not the username element is included in the regsitration document returned. This can be used in conjunction with match_by_username.

Type xsd:boolean.

Default is false.

secondary_participation_id

Optional. A comma-separated list of secondary TeamRaiser participation type IDs.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

participation_id

Required. The TeamRaiser participation type ID.

Type xsd:nonNegativeInteger.

include_username_element

Optional. Determines whether or not the username element is included in the regsitration document returned. This can be used in conjunction with match_by_username.

Type xsd:boolean.

Default is false.

secondary_participation_id

Optional. A comma-separated list of secondary TeamRaiser participation type IDs.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<processRegistrationRequest xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <eventId>1000</eventId>
  <teamId xsi:nil="true"/>
  <teamName xsi:nil="true"/>
  <teamGoal xsi:nil="true"/>
  <teamMembershipGoal xsi:nil="true"/>
  <teamPassword xsi:nil="true"/>
  <teamDivision xsi:nil="true"/>
```

```
  <companyId xsi:nil="true"/>
  <companyName xsi:nil="true"/>
  <paymentType xsi:nil="true"/>
  <checkNumber xsi:nil="true"/>
  <billingTitle xsi:nil="true"/>
  <billingFirstName xsi:nil="true"/>
  <billingMiddleName xsi:nil="true"/>
  <billingLastName xsi:nil="true"/>
  <billingSuffix xsi:nil="true"/>
  <billingEmail xsi:nil="true"/>
  <billingStreet1 xsi:nil="true"/>
  <billingStreet2 xsi:nil="true"/>
  <billingStreet3 xsi:nil="true"/>
  <billingCity xsi:nil="true"/>
  <billingState xsi:nil="true"/>
  <billingCounty xsi:nil="true"/>
  <billingZip xsi:nil="true"/>
  <billingCountry xsi:nil="true"/>
  <creditCardType xsi:nil="true"/>
  <creditCardNumber xsi:nil="true"/>
  <creditCardExpMonth xsi:nil="true"/>
  <creditCardExpYear xsi:nil="true"/>
  <creditCardVerificationCode xsi:nil="true"/>
  <batchId xsi:nil="true"/>
  <primaryRegistration>
    <partTypeId>1100</partTypeId>
    <consId xsi:nil="true"/>
    <memberId xsi:nil="true"/>
    <username xsi:nil="true"/>
    <promoCode xsi:nil="true"/>
    <discountCode xsi:nil="true"/>
    <fundraisingGoal xsi:nil="true"/>
    <acceptWaiver xsi:nil="true"/>
    <email required="true"/>
    <title xsi:nil="true"/>
    <firstName required="true"/>
    <middleName xsi:nil="true"/>
    <lastName required="true"/>
    <suffix xsi:nil="true"/>
    <professionalSuffix xsi:nil="true"/>
    <street1 xsi:nil="true"/>
    <street2 xsi:nil="true"/>
    <street3 xsi:nil="true"/>
    <city xsi:nil="true"/>
    <stateProv xsi:nil="true"/>
    <county xsi:nil="true"/>
    <postalCode xsi:nil="true"/>
    <country xsi:nil="true"/>
    <phoneNumber xsi:nil="true"/>
    <dateOfBirth xsi:nil="true"/>
    <gender xsi:nil="true"/>
    <employerName xsi:nil="true"/>
    <occupation xsi:nil="true"/>
    <anonymous xsi:nil="true"/>
    <screenName xsi:nil="true"/>
    <question>
      <id>1010</id>
      <type>PARTICIPATION_TYPE</type>
      <response xsi:nil="true"/>
      <key>T_SHIRT_SIZE</key>
      <hidden>false</hidden>
      <questionRequired>true</questionRequired>
    </question>
    <upsell>
```

```
      <id>1015</id>
      <optionId xsi:nil="true"/>
      <purchase>false</purchase>
    </upsell>
    <registrationFee xsi:nil="true"/>
    <additionalGift>
      <amount xsi:nil="true"/>
      <isPublic>true</isPublic>
      <isAnonymous>false</isAnonymous>
    </additionalGift>
    <emailOptIn>true</emailOptIn>
    <mailOptIn>false</mailOptIn>
    <emergencyContactName xsi:nil="true"/>
    <emergencyContactPhoneNumber xsi:nil="true"/>
    <honoreeTitle xsi:nil="true"/>
    <honoreeFirstName xsi:nil="true"/>
    <honoreeMiddleName xsi:nil="true"/>
    <honoreeLastName xsi:nil="true"/>
    <honoreeEmail xsi:nil="true"/>
    <honoreeStreet1 xsi:nil="true"/>
    <honoreeStreet2 xsi:nil="true"/>
    <honoreeStreet3 xsi:nil="true"/>
    <honoreeCity xsi:nil="true"/>
    <honoreeStateProv xsi:nil="true"/>
    <honoreeCounty xsi:nil="true"/>
    <honoreePostalCode xsi:nil="true"/>
    <honoreeCountry xsi:nil="true"/>
    <honoreeRelationship xsi:nil="true"/>
    <honoreeDeceased xsi:nil="true"/>
    <giftNotification xsi:nil="true"/>
    <emailFormat xsi:nil="true"/>
    <registrationDate xsi:nil="true"/>
    <checkInStatus xsi:nil="true"/>
    <flowStep xsi:nil="true"/>
  </primaryRegistration>
  <secondaryRegistration>
    <partTypeId>1100</partTypeId>
    <consId xsi:nil="true"/>
    <memberId xsi:nil="true"/>
    <username xsi:nil="true"/>
    <promoCode xsi:nil="true"/>
    <discountCode xsi:nil="true"/>
    <fundraisingGoal xsi:nil="true"/>
    <acceptWaiver xsi:nil="true"/>
    <email required="true"/>
    <title xsi:nil="true"/>
    <firstName required="true"/>
    <middleName xsi:nil="true"/>
    <lastName required="true"/>
    <suffix xsi:nil="true"/>
    <professionalSuffix xsi:nil="true"/>
    <street1 xsi:nil="true"/>
    <street2 xsi:nil="true"/>
    <street3 xsi:nil="true"/>
    <city xsi:nil="true"/>
    <stateProv xsi:nil="true"/>
    <county xsi:nil="true"/>
    <postalCode xsi:nil="true"/>
    <country xsi:nil="true"/>
    <phoneNumber xsi:nil="true"/>
    <dateOfBirth xsi:nil="true"/>
    <gender xsi:nil="true"/>
    <employerName xsi:nil="true"/>
    <occupation xsi:nil="true"/>
```

```
    <anonymous xsi:nil="true"/>
    <screenName xsi:nil="true"/>
    <question>
      <id>1010</id>
      <type>EVENT</type>
      <response xsi:nil="true"/>
      <key>Turned_In_Permission_Slip</key>
      <hidden>true</hidden>
      <questionRequired>true</questionRequired>
    </question>
    <upsell>
      <id>1015</id>
      <optionId xsi:nil="true"/>
      <purchase>true</purchase>
    </upsell>
    <registrationFee xsi:nil="true"/>
    <relationship xsi:nil="true"/>
  </secondaryRegistration>
</processRegistrationRequest>
```

JSON response

```
{"processRegistrationRequest":
{"teamName":null,"billingSuffix":null,"billingMiddleName":null,"billingEmail":null,"batch
{"dateOfBirth":null,"fundraisingGoal":null,"screenName":null,"registrationFee":null,"prof
{"optionId":null,"id":"1015","purchase":"true"},"question":
{"response":null,"id":"1010","questionRequired":"true","hidden":"true","type":"EVENT","ke
{"required":"true"},"discountCode":null,"promoCode":null,"middleName":null,"street2":nul
{"required":"true"},"street1":null,"occupation":null,"consId":null,"partTypeId":"1100","s
{"required":"true"},"county":null,"memberId":null,"street3":null},"primaryRegistration":
{"fundraisingGoal":null,"honoreeCity":null,"giftNotification":null,"screenName":null,"pro
{"required":"true"},"occupation":null,"consId":null,"partTypeId":"1100","emailFormat":nul
{"amount":null,"isPublic":"true","isAnonymous":"false"},"emailOptIn":"true","county":nul
{"required":"true"},"honoreeEmail":null,"memberId":null,"honoreePostalCode":null,"honoree
{"optionId":null,"id":"1015","purchase":"false"},"question":
{"response":null,"id":"1010","questionRequired":"true","hidden":"false","type":"PARTICIPA
{"required":"true"},"promoCode":null,"street2":null,"street1":null,"honoreeRelationship"
```

getSentMessage

Retrieves the message information for a specific sent message id.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getSentMessage` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&message_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

message_id

Required. Message ID

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getSentMessageResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <messageInfo>
    <messageId>123</messageId>
    <recipient>
      <contactId>1293</contactId>
      <firstName>example_first</firstName>
      <lastName>example_last</lastName>
      <email>name123@example.com</email>
    </recipient>
    <subject>Sample subject</subject>
    <messageBody>Sample message body.</messageBody>
    <messageName>Sample Name</messageName>
    <date>Thu, Jan 1 2009 at 1:00:00 PM</date>
  </messageInfo>
</getSentMessageResponse>
```

JSON response

```
{"getSentMessageResponse":{"messageInfo":{"messageBody":"Sample
 message body.","subject":"Sample subject","date":"Thu,
 Jan 1 2009 at 1:00:00 PM","messageId":"123","recipient":
{"lastName":"example_last","contactId":"1293","email":"name123@example.com","firstName":"
 Name"}}}
```

getSentMessages

Provides a list of sent messages for a participant

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getSentMessages`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&message_filters=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

message_filters

Optional. Message search filters

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getSentMessagesResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>1</totalNumberResults>
  <messageItem>
    <date>1/1/09</date>
    <message_id>1001</message_id>
    <recipients>name1@example.com</recipients>
    <subject>example subject</subject>
  </messageItem>
</getSentMessagesResponse>
```

JSON response

```
{"getSentMessagesResponse":{"totalNumberResults":"1","messageItem":
{"recipients":"name1@example.com","subject":"example
 subject","message_id":"1001","date":"1/1/09"}}}
```

getShortcut

Provides information about a participant's shortcut URL.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getShortcut`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [
`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*

]`[ &sub_source=`*value* `] [ &success_redirect=`*value* `] [ &suppress_response_codes=`*value* `]`
`&fr_id=`*value* `[ &auth=`*value* `] [ &sso_auth_token=`*value* `]`

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getShortcutResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <shortcutItem>
    <defaultUrl>http://example.com/personalpageurl</defaultUrl>
    <hint>Sample hint text.</hint>
    <prefix>http://example.com/goto/</prefix>
    <text>sampleuser</text>
    <url>http://example.com/goto/sampleuser</url>
  </shortcutItem>
</getShortcutResponse>
```

JSON response

```
{"getShortcutResponse":{"shortcutItem":{"text":"sampleuser","prefix":"http://
example.com/goto/","hint":"Sample hint text.","defaultUrl":"http://
example.com/personalpageurl","url":"http://example.com/goto/sampleuser"}}}
```

getSuggestedMessage

Provides detailed information about a suggested message.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getSuggestedMessage &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&message_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

message_id

Required. Message ID

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getSuggestedMessageResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <messageInfo>
    <messageId>123</messageId>
    <subject>Sample subject</subject>
    <messageBody>Sample message body.</messageBody>
    <messageName>Sample Name</messageName>
    <active>true</active>
    <messageType>THANKS</messageType>
  </messageInfo>
</getSuggestedMessageResponse>
```

JSON response

```
{"getSuggestedMessageResponse":{"messageInfo":
{"messageBody":"Sample message body.","subject":"Sample
 subject","active":"true","messageType":"THANKS","messageId":"123","messageName":"Sample
 Name"}}}
```

getSuggestedMessages

Provides a list of suggested messages for the participant center.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getSuggestedMessages &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getSuggestedMessagesResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <suggestedMessage>
    <messageId>100</messageId>
    <name>Sample Layout 01</name>
    <messageType>THANKS</messageType>
    <personal>false</personal>
    <active>true</active>
  </suggestedMessage>
  <suggestedMessage>
    <messageId>101</messageId>
    <name>Sample Layout 11</name>
    <messageType>THANKS</messageType>
    <personal>false</personal>
    <active>true</active>
  </suggestedMessage>
  <suggestedMessage>
    <messageId>102</messageId>
```

```
    <name>Sample Layout 21</name>
    <messageType>THANKS</messageType>
    <personal>false</personal>
    <active>true</active>
  </suggestedMessage>
  <suggestedMessage>
    <messageId>103</messageId>
    <name>Sample Layout 31</name>
    <messageType>THANKS</messageType>
    <personal>false</personal>
    <active>true</active>
  </suggestedMessage>
  <suggestedMessage>
    <messageId>104</messageId>
    <name>Sample Layout 41</name>
    <messageType>THANKS</messageType>
    <personal>false</personal>
    <active>true</active>
  </suggestedMessage>
</getSuggestedMessagesResponse>
```

JSON response

```
{"getSuggestedMessagesResponse":{"suggestedMessage":
[{"personal":"false","name":"Sample Layout
 01","active":"true","messageType":"THANKS","messageId":"100"},
{"personal":"false","name":"Sample Layout
 11","active":"true","messageType":"THANKS","messageId":"101"},
{"personal":"false","name":"Sample Layout
 21","active":"true","messageType":"THANKS","messageId":"102"},
{"personal":"false","name":"Sample Layout
 31","active":"true","messageType":"THANKS","messageId":"103"},
{"personal":"false","name":"Sample Layout
 41","active":"true","messageType":"THANKS","messageId":"104"}]}}
```

getSurveyResponses

Provides a list of survey responses.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getSurveyResponses` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&question_=`*value*] [`&requested_only=`*value*] [`&survey_id=`*value*] [`&use_filters=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

question_

Optional. Include either question_ or question_key_ parameters for each question on the survey. Provide either existing values returned by the getSurveyResponses method or updated values. For questions with IDs 123 and 456, include parameters question_123 and question_456.

Type xsd:string.

requested_only

Optional. This boolean value determines whether or not to include responses for questions in the survey you did not explicitly request.

Type xsd:boolean.

survey_id

Optional. The survey ID (default is the event registration survey).

Type xsd:nonNegativeInteger.

use_filters

Optional. Whether or not to filter the list of questions based on participation type or team captaincy. Default is False.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getSurveyResponsesResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <responses>
    <questionText>Sample question 1</questionText>
    <responseValue>SampleResponse</responseValue>
    <questionId>101</questionId>
    <surveyInstanceId>1111</surveyInstanceId>
    <userId>1234567</userId>
    <key>T-Shirt_Size</key>
    <isHidden>false</isHidden>
  </responses>
  <responses>
    <questionText>Sample question 2</questionText>
    <responseValue>SampleResponse</responseValue>
    <questionId>102</questionId>
    <surveyInstanceId>1111</surveyInstanceId>
    <userId>1234567</userId>
    <key>T-Shirt_Size</key>
    <isHidden>false</isHidden>
  </responses>
```

```
  <responses>
    <questionText>Sample question 3</questionText>
    <responseValue>SampleResponse</responseValue>
    <questionId>103</questionId>
    <surveyInstanceId>1111</surveyInstanceId>
    <userId>1234567</userId>
    <key>T-Shirt_Size</key>
    <isHidden>false</isHidden>
  </responses>
</getSurveyResponsesResponse>
```

JSON response

```
{"getSurveyResponsesResponse":{"responses":
[{"isHidden":"false","questionId":"101","userId":"1234567","questionText":"Sample
 question 1","surveyInstanceId":"1111","key":"T-
Shirt_Size","responseValue":"SampleResponse"},
{"isHidden":"false","questionId":"102","userId":"1234567","questionText":"Sample
 question 2","surveyInstanceId":"1111","key":"T-
Shirt_Size","responseValue":"SampleResponse"},
{"isHidden":"false","questionId":"103","userId":"1234567","questionText":"Sample
 question 3","surveyInstanceId":"1111","key":"T-
Shirt_Size","responseValue":"SampleResponse"}]}}
```

getTeam

Provides detailed information about a team.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeam &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&team_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

team_id

Required. A unique team ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <team>
    <name>Sample Team Name</name>
    <description>Sample Team Description</description>
    <captainsMessage>Sample Captains Message</captainsMessage>
    <goal>50000</goal>
    <id>12345</id>
  </team>
</getTeamResponse>
```

JSON response

```
{"getTeamResponse":{"team":{"id":"12345","description":"Sample Team
 Description","name":"Sample Team Name","captainsMessage":"Sample Captains
 Message","goal":"50000"}}}
```

getTeamCaptains

Provides a list of team captains.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamCaptains`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*
`&team_id=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

team_id

Required. A unique team ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamCaptainsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <captain>
    <name>
      <first>Cap</first>
      <last>Sample01</last>
    </name>
    <aTeamCaptain>true</aTeamCaptain>
    <teamName>Sample Team Name</teamName>
    <donationUrl>http://example.com/donateurl</donationUrl>
    <personalPageUrl>http://example.com/personalpageurl</personalPageUrl>
    <teamPageUrl>http://example.com/teampageurl</teamPageUrl>
    <consId>1001000</consId>
  </captain>
  <captain>
    <name>
      <first>Cap</first>
      <last>Sample11</last>
    </name>
    <aTeamCaptain>true</aTeamCaptain>
    <teamName>Sample Team Name</teamName>
    <donationUrl>http://example.com/donateurl</donationUrl>
    <personalPageUrl>http://example.com/personalpageurl</personalPageUrl>
    <teamPageUrl>http://example.com/teampageurl</teamPageUrl>
    <consId>1001001</consId>
  </captain>
  <captain>
    <name>
      <first>Cap</first>
      <last>Sample21</last>
    </name>
    <aTeamCaptain>true</aTeamCaptain>
    <teamName>Sample Team Name</teamName>
    <donationUrl>http://example.com/donateurl</donationUrl>
    <personalPageUrl>http://example.com/personalpageurl</personalPageUrl>
    <teamPageUrl>http://example.com/teampageurl</teamPageUrl>
    <consId>1001002</consId>
  </captain>
</getTeamCaptainsResponse>
```

JSON response

```
{"getTeamCaptainsResponse":{"captain":[{"teamName":"Sample Team
 Name","aTeamCaptain":"true","teamPageUrl":"http://example.com/
teampageurl","consId":"1001000","donationUrl":"http://example.com/
donateurl","name":{"last":"Sample01","first":"Cap"},"personalPageUrl":"http://
example.com/personalpageurl"},{"teamName":"Sample Team
 Name","aTeamCaptain":"true","teamPageUrl":"http://example.com/
teampageurl","consId":"1001001","donationUrl":"http://example.com/
donateurl","name":{"last":"Sample11","first":"Cap"},"personalPageUrl":"http://
example.com/personalpageurl"},{"teamName":"Sample Team
 Name","aTeamCaptain":"true","teamPageUrl":"http://example.com/
```

```
teampageurl","consId":"1001002","donationUrl":"http://example.com/
donateurl","name":{"last":"Sample21","first":"Cap"},"personalPageUrl":"http://
example.com/personalpageurl"}]}}
```

getTeamDivisions

Returns a list of team divisions for a teamraiser.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamDivisions &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamDivisionsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <divisionName>sample division name</divisionName>
</getTeamDivisionsResponse>
```

JSON response

```
{"getTeamDivisionsResponse":{"divisionName":"sample division name"}}
```

getTeamDonationByDay

Provides data used to build daily progress charts.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamDonationByDay &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamDonationByDayResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <donationDataByDay>
    <date>2013-02-02-06:00</date>
    <amount>4</amount>
  </donationDataByDay>
  <donationDataByDay>
    <date>2013-02-03-06:00</date>
    <amount>8</amount>
  </donationDataByDay>
</getTeamDonationByDayResponse>
```

JSON response

```
{"getTeamDonationByDayResponse":{"donationDataByDay":
[{"amount":"4","date":"2013-02-02-06:00"},
{"amount":"8","date":"2013-02-03-06:00"}]}}
```

getTeamGifts

Provides a list of gifts.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamGifts`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [
`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*

]`[ &sub_source=`*value* `]`[`&success_redirect=`*value* `]`[`&suppress_response_codes=`*value* `] &cons_id=`*value* `&fr_id=`*value* `[ &list_ascending=`*value* `]`[`&list_category_id=`*value* `]`[`&list_filter_column=`*value* `]`[`&list_filter_text=`*value* `]`[`&list_page_offset=`*value* `]`[`&list_page_size=`*value* `]`[`&list_record_offset=`*value* `]`[`&list_sort_column=`*value* `]`[`&auth=`*value* `]`[`&sso_auth_token=`*value* `]`

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Required. A unique constituent ID.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getGiftsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>2</totalNumberResults>
  <gift>
    <acknowledged>false</acknowledged>
    <checkNumber xsi:nil="true"/>
    <confirmed>true</confirmed>
    <date>2013-02-03T05:18:26.632-06:00</date>
    <email>name1@example.com</email>
    <giftAmount>2501</giftAmount>
    <giftAidStatus>1</giftAidStatus>
    <giftMessage>Sample gift message.</giftMessage>
    <id>12001</id>
    <isRecurring>false</isRecurring>
    <name>
      <first>Jo</first>
      <middle>1</middle>
      <last>Public</last>
    </name>
    <paymentType>CREDIT</paymentType>
    <recognitionName>Sample recognition name 1</recognitionName>
  </gift>
  <gift>
    <acknowledged>false</acknowledged>
    <checkNumber xsi:nil="true"/>
    <confirmed>true</confirmed>
    <date>2013-02-03T05:18:26.632-06:00</date>
    <email>name1@example.com</email>
    <giftAmount>2502</giftAmount>
    <giftAidStatus>1</giftAidStatus>
    <giftMessage>Sample gift message.</giftMessage>
    <id>12002</id>
    <isRecurring>false</isRecurring>
    <name>
      <first>Jo</first>
      <middle>2</middle>
      <last>Public</last>
    </name>
    <paymentType>CREDIT</paymentType>
    <recognitionName>Sample recognition name 2</recognitionName>
  </gift>
```

```
</getGiftsResponse>
```

JSON response

```
{"getGiftsResponse":{"gift":[{"giftMessage":"Sample gift
 message.","isRecurring":"false","paymentType":"CREDIT","checkNumber":null,"date":"2013-0
 recognition name
 1","confirmed":"true","email":"name1@example.com","giftAidStatus":"1","acknowledged":"fa
{"last":"Public","middle":"1","first":"Jo"},"giftAmount":"2501"},
{"giftMessage":"Sample gift
 message.","isRecurring":"false","paymentType":"CREDIT","checkNumber":null,"date":"2013-0
 recognition name
 2","confirmed":"true","email":"name1@example.com","giftAidStatus":"1","acknowledged":"fa
{"last":"Public","middle":"2","first":"Jo"},"giftAmount":"2502"}],"totalNumberResults":"2
```

getTeamMembers

Provides a list of team members.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamMembers &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&team_id=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

team_id

Required. A unique team ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamMembersResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <member>
    <name>
      <first>Member</first>
      <last>Sample01</last>
    </name>
    <aTeamCaptain>true</aTeamCaptain>
```

```
    <teamName>Sample Team Name</teamName>
    <donationUrl>http://example.com/donateurl</donationUrl>
    <personalPageUrl>http://example.com/personalpageurl</personalPageUrl>
    <teamPageUrl>http://example.com/teampageurl</teamPageUrl>
    <consId>1001000</consId>
  </member>
  <member>
    <name>
      <first>Member</first>
      <last>Sample11</last>
    </name>
    <aTeamCaptain>false</aTeamCaptain>
    <teamName>Sample Team Name</teamName>
    <donationUrl>http://example.com/donateurl</donationUrl>
    <personalPageUrl>http://example.com/personalpageurl</personalPageUrl>
    <teamPageUrl>http://example.com/teampageurl</teamPageUrl>
    <consId>1001001</consId>
  </member>
  <member>
    <name>
      <first>Member</first>
      <last>Sample21</last>
    </name>
    <aTeamCaptain>false</aTeamCaptain>
    <teamName>Sample Team Name</teamName>
    <donationUrl>http://example.com/donateurl</donationUrl>
    <personalPageUrl>http://example.com/personalpageurl</personalPageUrl>
    <teamPageUrl>http://example.com/teampageurl</teamPageUrl>
    <consId>1001002</consId>
  </member>
</getTeamMembersResponse>
```

JSON response

```
{"getTeamMembersResponse":{"member":[{"teamName":"Sample Team
 Name","aTeamCaptain":"true","teamPageUrl":"http://example.com/
teampageurl","consId":"1001000","donationUrl":"http://example.com/
donateurl","name":
{"last":"Sample01","first":"Member"},"personalPageUrl":"http://
example.com/personalpageurl"},{"teamName":"Sample Team
 Name","aTeamCaptain":"false","teamPageUrl":"http://example.com/
teampageurl","consId":"1001001","donationUrl":"http://example.com/
donateurl","name":
{"last":"Sample11","first":"Member"},"personalPageUrl":"http://
example.com/personalpageurl"},{"teamName":"Sample Team
 Name","aTeamCaptain":"false","teamPageUrl":"http://example.com/
teampageurl","consId":"1001002","donationUrl":"http://example.com/
donateurl","name":
{"last":"Sample21","first":"Member"},"personalPageUrl":"http://example.com/
personalpageurl"}]}}
```

getTeamPageInfo

Provides information used to display or edit a team page.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamPageInfo`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamPageResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <teamPage>
    <description>Sample Team Page Description</description>
    <richText>&lt;P>Sample Team Page rich text.&lt;/P></richText>
    <richTextEnabled>true</richTextEnabled>
  </teamPage>
</getTeamPageResponse>
```

JSON response

```
{"getTeamPageResponse":{"teamPage":{"description":"Sample Team
 Page Description","richText":"<P>Sample Team Page rich text.<\/
P>","richTextEnabled":"true"}}}
```

getTeamPagePreview

Retrieves a preview of the team page.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamPagePreview &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&description=`*value*] [`&rich_text=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method requires authentication. The logged-in user must be a Team Captain to execute this method successfully.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

description

Optional. Description

Type xsd:string.

rich_text

Optional. The rich text area of the page.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamPagePreviewResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <previewUrl>http://www.example.com/preview.html</previewUrl>
</getTeamPagePreviewResponse>
```

JSON response

```
{"getTeamPagePreviewResponse":{"previewUrl":"http://www.example.com/
preview.html"}}
```

getTeamPhoto

Retrieves the team photo for a team.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamPhoto`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method requires authentication. The logged-in user must be a Team Captain to execute this method successfully.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamPhotoResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <photoItem>
    <id>1</id>
    <thumbnailUrl>&lt;url></thumbnailUrl>
    <customUrl>&lt;url></customUrl>
    <originalUrl>&lt;url></originalUrl>
  </photoItem>
</getTeamPhotoResponse>
```

JSON response

```
{"getTeamPhotoResponse":{"photoItem":
{"id":"1","originalUrl":"<url>","thumbnailUrl":"<url>","customUrl":"<url>"}}}
```

getTeamRoster

Returns the team roster for a given team id

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamRoster`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* `&team_id=`*value* [`&include_download_url=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&positive_amount_only=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

team_id

Required. A unique team ID

Type xsd:nonNegativeInteger.

include_download_url

Optional. Specifies whether a url to download the data should be included in the response.

Type xsd:boolean.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

positive_amount_only

Optional. Setting to only return team members with a positive amount if set to REQUIRED.

Type xsd:boolean.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamRosterResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <teamMember>
    <firstName>first1</firstName>
    <lastName>lastname</lastName>
    <email>example1@example.com</email>
    <amount>1001</amount>
    <aTeamCaptain>false</aTeamCaptain>
    <consId>1001001</consId>
  </teamMember>
  <teamMember>
    <firstName>first2</firstName>
    <lastName>lastname</lastName>
    <email>example2@example.com</email>
    <amount>1002</amount>
```

```
    <aTeamCaptain>false</aTeamCaptain>
    <consId>1001002</consId>
  </teamMember>
  <teamMember>
    <firstName>first3</firstName>
    <lastName>lastname</lastName>
    <email>example3@example.com</email>
    <amount>1003</amount>
    <aTeamCaptain>false</aTeamCaptain>
    <consId>1001003</consId>
  </teamMember>
</getTeamRosterResponse>
```

JSON response

```
{"getTeamRosterResponse":{"teamMember":
[{"amount":"1001","aTeamCaptain":"false","lastName":"lastname","consId":"1001001","email"
{"amount":"1002","aTeamCaptain":"false","lastName":"lastname","consId":"1001002","email"
{"amount":"1003","aTeamCaptain":"false","lastName":"lastname","consId":"1001003","email"
```

getTeamsByInfo

Provides detailed information about a team.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	SRTeamraiserAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamsByInfo`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] [`&event_type=`*value*] [`&event_type2=`*value*] [`&event_type3=`*value*] [`&first_name=`*value*] [`&fr_id=`*value*] [`&full_search=`*value*] [`&last_name=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&public_event_type=`*value*] [`&screenname=`*value*] [`&team_company=`*value*] [`&team_company_id=`*value*] [`&team_division=`*value*] [`&team_id=`*value*] [`&team_name=`*value*]

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRTeamraiserAPI?method=getTeamsByInfo`
`&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] [`&event_type=`*value*] [`&event_type2=`*value*] [`&event_type3=`*value*] [`&first_name=`*value*] [`&fr_id=`*value*] [`&full_search=`*value*] [`&last_name=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&public_event_type=`*value*] [`&screenname=`*value*] [`&team_company=`*value*] [`&team_company_id=`*value*] [`&team_division=`*value*] [`&team_id=`*value*] [`&team_name=`*value*]

Usage Notes

This method returns a list of active TeamRaiser Teams, allowing you to retrieve information on a single team specified by team_id, or to retrieve a list of teams by filtering using other parameters.

If you do not supply the team_id, you must include at least one of the following filter parameters:

- company_name
- company_id
- event_type
- event_type2
- event_type3
- fr_id
- team_divsion

Filters are conjoined using AND logic; the API returns the intersection of the filtered result sets.

Note: You must supply fr_id in order to retrieve performance data (amountRaised).

This method returns data for teams associated with events that are Unpublished, Accepting Registrations Only, Accepting Regisrations and Gifts, Accepting Gifts Only, or Closed. It does not return data for teams associated with Deleted or Archived events. If the associated event is not accepting registrations, the `joinTeamURL` element of the response document will be null. If the event is not accepting gifts, the `teamDonateURL` element will be null. Both elements will be null for Unpublshed or Closed events.

Note: When searching on company, team, or team captain name, the search is a "contains" search, and is case insensitive. A search for `first_name=jo` will return participants with a First name of "Jon", "John", "Joan", etc.

To search by team captain, you must include a certain number of characters in order for the system to be able to perform the search. The default minimum characters comes from the Site Data Parameter `F2F_PART_FINDER_MIN_CRITERIA`. If that SDP is not set to a number, the search defaults to requiring a minimum of 3 characters. This minimum number of characters is applied to a combination of the first_name and last_name parameters.

Note: If the minimum characters is 3, then you must provide at least 3 characters total between the first_name and last_name parameters. A search for `first_name=n&last_name=ng` would be successful, but a search for `first_name=n&last_name=n` would generate an error.

Additionally, the list of Teams that is returned supports several parameters allowing finer control of the results, including additional filtering, pagination, and sorting.

Paginating the List takes two parameters: list_page_size, which specifies the number of results to be retrieved for that page. You must also provide **one** of the following two parameters:

- list_page_offset: specifies which page of the results to retrieve. The first page of results has a list_page_offset of 0, page 2 has a list_page_offset of 1, and so on.
- list_record_offset: specifies which record the page should start with. This also uses numbering from 0, so the first record has a list_record_offset of 0, the 2nd record has a list_record_offset of 1, and so on.

Filtering the List takes two parameters:

- list_filter_column: name of the element to be filtered on.
- list_filter_text: text that the element specified by list_filter_column must contain.

Note: list_filter_text performs an exact match, and is case sensitive.

Sorting the List takes two parameters:

- list_sort_column: name of the element to be sorted by.
- list_ascending: Boolean value indicating whether the sort should be in ascending order

Note: You can paginate, filter, and sort the list all in one call.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

event_type

Optional. The name of the Primary Event Type to limit search results to. Omit this parameter to search on all Primary Event Types. Defaults to null.

Type xsd:string.

event_type2

Optional. The name of the Event Type 2 to limit search results to. Omit this parameter to search on events with any Event Type 2. Defaults to null.

Type xsd:string.

event_type3

Optional. The name of the Event Type 3 to limit search results to. Omit this parameter to search on events with any Event Type 3. Defaults to null.

Type xsd:string.

first_name

Optional. Team Captain first name.

Type xsd:string.

fr_id

Optional. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

full_search

Optional. If TRUE, returns all events regardless of event status, otherwise returns only published events. Possible status values include: Unpublished = 0, Accepting Registrations Only = 1, AcceptingRegistrations and Gifts = 2, Accepting Gifts Only= 3, Closed = 4, Archived = 8, Deleted = 9.

Type xsd:boolean.

last_name

Optional. Team Captain last name.

Type xsd:string.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

public_event_type

Optional. The case-sensitive name of the Public Event Type to limit search results to. Omit this parameter to search on events with any Public Event Type. Defaults to null.

Type xsd:string.

screenname

Optional. Team Captain screen name.

Type xsd:string.

team_company

Optional. The name of the team company.

Type xsd:string.

team_company_id

Optional. The id of the team company.

Type xsd:nonNegativeInteger.

team_division

Optional. The team division.

Type xsd:string.

team_id

Optional. A unique team ID.

Type xsd:nonNegativeInteger.

team_name

Optional. A unique team name.

Type xsd:string.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

event_type

Optional. The name of the Primary Event Type to limit search results to. Omit this parameter to search on all Primary Event Types. Defaults to null.

Type xsd:string.

event_type2

Optional. The name of the Event Type 2 to limit search results to. Omit this parameter to search on events with any Event Type 2. Defaults to null.

Type xsd:string.

event_type3

Optional. The name of the Event Type 3 to limit search results to. Omit this parameter to search on events with any Event Type 3. Defaults to null.

Type xsd:string.

first_name

Optional. Team Captain first name.

Type xsd:string.

fr_id

Optional. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

full_search

Optional. If TRUE, returns all events regardless of event status, otherwise returns only published events. Possible status values include: Unpublished = 0, Accepting Registrations Only = 1, AcceptingRegistrations and Gifts = 2, Accepting Gifts Only= 3, Closed = 4, Archived = 8, Deleted = 9.

Type xsd:boolean.

last_name

Optional. Team Captain last name.

Type xsd:string.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

public_event_type

Optional. The case-sensitive name of the Public Event Type to limit search results to. Omit this parameter to search on events with any Public Event Type. Defaults to null.

Type xsd:string.

screenname

Optional. Team Captain screen name.

Type xsd:string.

team_company

Optional. The name of the team company.

Type xsd:string.

team_company_id

Optional. The id of the team company.

Type xsd:nonNegativeInteger.

team_division

Optional. The team division.

Type xsd:string.

team_id

Optional. A unique team ID.

Type xsd:nonNegativeInteger.

team_name

Optional. A unique team name.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamSearchByInfoResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>1</totalNumberResults>
  <team>
    <name>Sample Team Name</name>
    <amountRaised>15850</amountRaised>
    <goal>50000</goal>
    <id>12345</id>
    <captainConsId>1001001</captainConsId>
    <captainFirstName>Earl</captainFirstName>
    <captainLastName>Campbell</captainLastName>
    <captainScreenname>The Tyler Rose</captainScreenname>
    <isCaptainAnon>false</isCaptainAnon>
    <companyId>1000</companyId>
    <companyName>Convio</companyName>
    <numMembers>7</numMembers>
    <divisionName>South Central</divisionName>
    <requiresPassword>false</requiresPassword>
    <isHonoraryTeam>false</isHonoraryTeam>
    <originalTeamId>12345</originalTeamId>
    <previousTeamId>0</previousTeamId>
    <redirectShortcutURL>http://convio.com/goto/myTeam</redirectShortcutURL>
    <joinTeamURL>http://convio.com/site/TRR/ABC/General/2053821550?
pg=tfind&amp;fr_id=1000&amp;fr_tjoin=1003</joinTeamURL>
    <teamDonateURL xsi:nil="true"/>
    <teamEcommerceURL xsi:nil="true"/>
    <EventId>1000</EventId>
    <eventName>Austin fun run</eventName>
    <teamPageURL>http://convio.com/site/TRR/ABC/General/2053821550?
pg=team&amp;fr_id=1000&amp;team_id=1003</teamPageURL>
    <maxTeamSize>0</maxTeamSize>
  </team>
</getTeamSearchByInfoResponse>
```

JSON response

```
{"getTeamSearchByInfoResponse":{"totalNumberResults":"1","team":
{"joinTeamURL":"http://convio.com/site/TRR/ABC/General/2053821550?
pg=tfind&fr_id=1000&fr_tjoin=1003","teamEcommerceURL":null,"EventId":"1000","captainLastN
 Team
 Name","isHonoraryTeam":"false","previousTeamId":"0","maxTeamSize":"0","captainFirstName"
 Tyler
 Rose","teamDonateURL":null,"isCaptainAnon":"false","goal":"50000","requiresPassword":"fa
 Central","originalTeamId":"12345","numMembers":"7","redirectShortcutURL":"http://
convio.com/goto/myTeam","amountRaised":"15850","eventName":"Austin fun
 run","companyId":"1000","teamPageURL":"http://convio.com/site/TRR/ABC/
General/2053821550?pg=team&fr_id=1000&team_id=1003"}}}
```

Examples

To perform an search for all teams with a team captain named "Jane" and the company name "Convio" in any event:

```
https://secure2.convio.net/myorg/site/CRTeamraiserAPI?
method=getTeamsByInfo&api_key=MyAPIKey&v=1.0&first_name=Jane&company_name=Convio
```

To search the event with fr_id=1000, looking for teams with a first name containing "J", a last name containing "Doe", and to retrieve the first page of results, displaying 10 results per page:

```
https://secure2.convio.net/myorg/site/CRTeamraiserAPI?
method=getTeamsByInfo&api_key=MyAPIKey&v=1.0&fr_id=1000&team_name=j&list_page_size=10&lis
```

getTeamShortcut

Provides information about a team's shortcut URL.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamShortcut` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method requires authentication. The logged-in user must be a Team Captain to execute this method successfully.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamShortcutResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <shortcutItem>
```

```
    <defaultUrl>http://example.com/personalpageurl</defaultUrl>
    <hint>Sample hint text.</hint>
    <prefix>http://example.com/goto/</prefix>
    <text>sampleuser</text>
    <url>http://example.com/goto/sampleuser</url>
  </shortcutItem>
</getTeamShortcutResponse>
```

JSON response

```
{"getTeamShortcutResponse":{"shortcutItem":
{"text":"sampleuser","prefix":"http://example.com/goto/","hint":"Sample hint
 text.","defaultUrl":"http://example.com/personalpageurl","url":"http://
example.com/goto/sampleuser"}}}
```

getTeamraiserAddressBookContact

Provides a list of AddressBook contacts with additional Teamraiser specific information.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraiserAddressBookContact &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&contact_id=`*value* `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

contact_id

Required. A unique ID to identify an address book contact.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamraiserAddressBookContactResponse xsi:schemaLocation="http://
convio.com/crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd"
 xmlns="http://convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <addressBookContact>
    <id>101</id>
    <firstName>Sam</firstName>
    <lastName>Ple1</lastName>
    <email>sample1@example.com</email>
    <street1>1234 Some St.</street1>
    <street2>#42</street2>
    <street3/>
    <city>Austin</city>
    <county>Travis</county>
    <state>TX</state>
    <zip>78758</zip>
    <phone>(512) 555 1234</phone>
    <country>USA</country>
    <amountRaised>13501</amountRaised>
    <messagesSent>5</messagesSent>
    <messagesOpened>3</messagesOpened>
    <clickThroughs>1</clickThroughs>
    <group>
      <name>Group 1</name>
      <id>201</id>
    </group>
    <group>
      <name>Group 2</name>
      <id>202</id>
    </group>
    <group>
      <name>Group 3</name>
      <id>203</id>
    </group>
  </addressBookContact>
</getTeamraiserAddressBookContactResponse>
```

JSON response

```
{"getTeamraiserAddressBookContactResponse":{"addressBookContact":
{"street2":"#42","lastName":"Ple1","street1":"1234 Some
 St.","zip":"78758","phone":"(512) 555
 1234","state":"TX","messagesSent":"5","city":"Austin","country":"USA","id":"101","click
[{"id":"201","name":"Group 1"},{"id":"202","name":"Group 2"},
{"id":"203","name":"Group 3"}],"firstName":"Sam","street3":{}}}}
```

getTeamraiserAddressBookContacts

Provides a list of AddressBook contacts with additional Teamraiser specific information.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraiserAddressBookContacts &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&skip_groups=`*value*] [`&tr_ab_filter=`*email_rpt_show_all* | *email_rpt_show_never_emailed* | *email_rpt_show_donors* | *email_rpt_show_unthanked_donors* | *email_rpt_show_nondonors* | *email_rpt_show_nondonors_followup* | *email_rpt_show_lybunt_donors* | *email_rpt_show_ly_donors* | *email_rpt_show_teammates* | *email_rpt_show_nonteammates* | *email_rpt_show_lybunt_teammates* | *email_rpt_show_ly_teammates* | *email_rpt_show_company_coordinator_captains* | *email_rpt_show_company_coordinator_participants*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

To combine contacts which meet different filter criteria in the response, concatenate the filter terms passed to the tr_ab_filter parameter in a list separated by semi-colons. In addition to the listed filter types, you can also retrieve contacts who are members of a specific address book group by including the filter type email_rpt_group_ suffixed by the numeric ID of the address book group.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

skip_groups

Optional. Skips the retrieval of groups per contact if set to TRUE.

Type xsd:boolean.

Default is false.

tr_ab_filter

Optional. Filters address book entries by the given criteria.

Type xsd:string.

Default is email_rpt_show_all.

Options are:

- email_rpt_show_all
- email_rpt_show_never_emailed
- email_rpt_show_donors
- email_rpt_show_unthanked_donors
- email_rpt_show_nondonors
- email_rpt_show_nondonors_followup
- email_rpt_show_lybunt_donors
- email_rpt_show_ly_donors
- email_rpt_show_teammates
- email_rpt_show_nonteammates
- email_rpt_show_lybunt_teammates
- email_rpt_show_ly_teammates
- email_rpt_show_company_coordinator_captains
- email_rpt_show_company_coordinator_participants

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<getTeamraiserAddressBookContactsResponse xsi:schemaLocation="http://
convio.com/crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd"
 xmlns="http://convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <totalNumberResults>3</totalNumberResults>
  <addressBookContact>
    <id>101</id>
    <firstName>Sam</firstName>
    <lastName>Ple1</lastName>
    <email>sample1@example.com</email>
    <street1>1234 Some St.</street1>
    <street2>#42</street2>
    <street3/>
    <city>Austin</city>
    <county>Travis</county>
    <state>TX</state>
    <zip>78758</zip>
    <phone>(512) 555 1234</phone>
    <country>USA</country>
    <amountRaised>10700</amountRaised>
    <previousAmountRaised>10333</previousAmountRaised>
    <messagesSent>5</messagesSent>
    <messagesOpened>3</messagesOpened>
    <clickThroughs>1</clickThroughs>
  </addressBookContact>
  <addressBookContact>
    <id>102</id>
    <firstName>Sam</firstName>
    <lastName>Ple2</lastName>
    <email>sample2@example.com</email>
    <street1>1234 Some St.</street1>
    <street2>#42</street2>
    <street3/>
    <city>Austin</city>
    <county>Travis</county>
    <state>TX</state>
    <zip>78758</zip>
    <phone>(512) 555 1234</phone>
    <country>USA</country>
    <amountRaised>8910</amountRaised>
    <previousAmountRaised>8241</previousAmountRaised>
    <messagesSent>5</messagesSent>
    <messagesOpened>3</messagesOpened>
    <clickThroughs>1</clickThroughs>
    <group>
      <name>Group 1</name>
      <id>201</id>
    </group>
  </addressBookContact>
  <addressBookContact>
    <id>103</id>
    <firstName>Sam</firstName>
    <lastName>Ple3</lastName>
    <email>sample3@example.com</email>
    <street1>1234 Some St.</street1>
    <street2>#42</street2>
    <street3/>
    <city>Austin</city>
    <county>Travis</county>
    <state>TX</state>
    <zip>78758</zip>
    <phone>(512) 555 1234</phone>
    <country>USA</country>
    <amountRaised>8809</amountRaised>
```

```
    <previousAmountRaised>12380</previousAmountRaised>
    <messagesSent>5</messagesSent>
    <messagesOpened>3</messagesOpened>
    <clickThroughs>1</clickThroughs>
    <group>
      <name>Group 1</name>
      <id>201</id>
    </group>
    <group>
      <name>Group 2</name>
      <id>202</id>
    </group>
  </addressBookContact>
</getTeamraiserAddressBookContactsResponse>
```

JSON response

```
{"getTeamraiserAddressBookContactsResponse":
{"totalNumberResults":"3","addressBookContact":
[{"street2":"#42","lastName":"Ple1","street1":"1234 Some
 St.","zip":"78758","phone":"(512) 555
 1234","state":"TX","messagesSent":"5","city":"Austin","country":"USA","id":"101","click
{}},{"street2":"#42","lastName":"Ple2","street1":"1234 Some
 St.","zip":"78758","phone":"(512) 555
 1234","state":"TX","messagesSent":"5","city":"Austin","country":"USA","id":"102","click
{"id":"201","name":"Group
 1"},"firstName":"Sam","previousAmountRaised":"5055","street3":
{}},{"street2":"#42","lastName":"Ple3","street1":"1234 Some
 St.","zip":"78758","phone":"(512) 555
 1234","state":"TX","messagesSent":"5","city":"Austin","country":"USA","id":"103","click
[{"id":"201","name":"Group 1"},{"id":"202","name":"Group
 2"}],"firstName":"Sam","previousAmountRaised":"11560","street3":{}}]}}
```

getTeamraiserAddressBookContactsByIds

Returns a list of contacts for a given list of contact ids.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraiserAddressBookContactsByIds &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&contact_ids=`*value* `&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

contact_ids

Required. A comma-separated list of ids to retrieve.

Type xsd:string.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamraiserAddressBookContactsByIdsResponse xsi:schemaLocation="http://
convio.com/crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd"
 xmlns="http://convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <addressBookContact>
    <id>1123</id>
    <firstName>firstName</firstName>
    <lastName>lastName</lastName>
    <email>example+1123@example.com</email>
  </addressBookContact>
</getTeamraiserAddressBookContactsByIdsResponse>
```

JSON response

```
{"getTeamraiserAddressBookContactsByIdsResponse":{"addressBookContact":
{"id":"1123","lastName":"lastName","email":"example
+1123@example.com","firstName":"firstName"}}}
```

getTeamraiserAddressBookContactsGiftHistory

Returns gift history for a participant's address book contacts aggregated by contact and event.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraiserAddressBookContactsGiftHistory`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* `&contact_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

contact_id

Required. The system-generated unique contact ID for the contact.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamraiserAddressBookContactsGiftHistoryResponse
 xsi:schemaLocation="http://convio.com/crm/v1.0 http://service.convio.net/
xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/v1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <eventContactGiftHistory>
    <fr_id>1000</fr_id>
    <internal_event_name>Internal Event Name</internal_event_name>
    <external_event_name>External Event Name</external_event_name>
    <primary_classification>Primary Classification</primary_classification>
    <event_date>2013-02-03T05:18:26.588-06:00</event_date>
    <contact_id>1001</contact_id>
    <amount>250.0</amount>
  </eventContactGiftHistory>
</getTeamraiserAddressBookContactsGiftHistoryResponse>
```

JSON response

```
{"getTeamraiserAddressBookContactsGiftHistoryResponse":
{"eventContactGiftHistory":
{"amount":"250.0","fr_id":"1000","contact_id":"1001","event_date":"2013-02-03T05:18:26.58
 Event Name","external_event_name":"External Event
 Name","primary_classification":"Primary Classification"}}}
```

getTeamraiserAddressBookFilters

Provides a list of event appropriate filters to be used in combination with the getTeamraiserAddressBookContacts method.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraiserAddressBookFilters &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&cons_id=`*value* `&fr_id=`*value* [`&include_past_teammates_filters=`*value*] [`&include_returning_team_filters=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

cons_id

Required. A unique constituent ID.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

include_past_teammates_filters

Optional. Include Past Teammates filters in the list of address book filters available

Type xsd:boolean.

Default is false.

include_returning_team_filters

Optional. Include Returing Team filters in the list of address book filters available.

Type xsd:boolean.

Default is false.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamraiserAddressBookFiltersResponse xsi:schemaLocation="http://
convio.com/crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd"
 xmlns="http://convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <filterGroup>
    <groupName>Test Group 1</groupName>
    <filter>
      <filterName>Test Filter 1</filterName>
      <filterValue>sample_value_1</filterValue>
      <teamName xsi:nil="true"/>
      <contactsCount>1</contactsCount>
    </filter>
    <filter>
      <filterName>Test Filter 2</filterName>
      <filterValue>sample_value_2</filterValue>
      <teamName xsi:nil="true"/>
      <contactsCount>2</contactsCount>
    </filter>
    <filter>
      <filterName>Test Filter 3</filterName>
      <filterValue>sample_value_3</filterValue>
```

```
      <teamName xsi:nil="true"/>
      <contactsCount>3</contactsCount>
    </filter>
    <filter>
      <filterName>Test Filter 4</filterName>
      <filterValue>sample_value_4</filterValue>
      <teamName xsi:nil="true"/>
      <contactsCount>4</contactsCount>
    </filter>
  </filterGroup>
  <filterGroup>
    <groupName>Test Group 2</groupName>
    <filter>
      <filterName>Test Filter 1</filterName>
      <filterValue>sample_value_1</filterValue>
      <teamName xsi:nil="true"/>
      <contactsCount>2</contactsCount>
    </filter>
    <filter>
      <filterName>Test Filter 2</filterName>
      <filterValue>sample_value_2</filterValue>
      <teamName xsi:nil="true"/>
      <contactsCount>4</contactsCount>
    </filter>
    <filter>
      <filterName>Test Filter 3</filterName>
      <filterValue>sample_value_3</filterValue>
      <teamName xsi:nil="true"/>
      <contactsCount>6</contactsCount>
    </filter>
    <filter>
      <filterName>Test Filter 4</filterName>
      <filterValue>sample_value_4</filterValue>
      <teamName xsi:nil="true"/>
      <contactsCount>8</contactsCount>
    </filter>
  </filterGroup>
  <filterGroup>
    <groupName>Test Group 3</groupName>
    <filter>
      <filterName>Test Filter 1</filterName>
      <filterValue>sample_value_1</filterValue>
      <teamName xsi:nil="true"/>
      <contactsCount>3</contactsCount>
    </filter>
    <filter>
      <filterName>Test Filter 2</filterName>
      <filterValue>sample_value_2</filterValue>
      <teamName xsi:nil="true"/>
      <contactsCount>6</contactsCount>
    </filter>
    <filter>
      <filterName>Test Filter 3</filterName>
      <filterValue>sample_value_3</filterValue>
      <teamName xsi:nil="true"/>
      <contactsCount>9</contactsCount>
    </filter>
    <filter>
      <filterName>Test Filter 4</filterName>
      <filterValue>sample_value_4</filterValue>
      <teamName xsi:nil="true"/>
      <contactsCount>12</contactsCount>
    </filter>
  </filterGroup>
```

```
</getTeamraiserAddressBookFiltersResponse>
```

JSON response


```
{"getTeamraiserAddressBookFiltersResponse":{"filterGroup":[{"groupName":"Test
 Group 1","filter":
[{"teamName":null,"filterValue":"sample_value_1","filterName":"Test Filter
 1","contactsCount":"1"},
{"teamName":null,"filterValue":"sample_value_2","filterName":"Test Filter
 2","contactsCount":"2"},
{"teamName":null,"filterValue":"sample_value_3","filterName":"Test Filter
 3","contactsCount":"3"},
{"teamName":null,"filterValue":"sample_value_4","filterName":"Test
 Filter 4","contactsCount":"4"}]},{"groupName":"Test Group 2","filter":
[{"teamName":null,"filterValue":"sample_value_1","filterName":"Test Filter
 1","contactsCount":"2"},
{"teamName":null,"filterValue":"sample_value_2","filterName":"Test Filter
 2","contactsCount":"4"},
{"teamName":null,"filterValue":"sample_value_3","filterName":"Test Filter
 3","contactsCount":"6"},
{"teamName":null,"filterValue":"sample_value_4","filterName":"Test
 Filter 4","contactsCount":"8"}]},{"groupName":"Test Group 3","filter":
[{"teamName":null,"filterValue":"sample_value_1","filterName":"Test Filter
 1","contactsCount":"3"},
{"teamName":null,"filterValue":"sample_value_2","filterName":"Test Filter
 2","contactsCount":"6"},
{"teamName":null,"filterValue":"sample_value_3","filterName":"Test Filter
 3","contactsCount":"9"},
{"teamName":null,"filterValue":"sample_value_4","filterName":"Test Filter
 4","contactsCount":"12"}]}]}}
```


getTeamraiserAddressBookGroupContacts

Returns a list of contacts for a given group.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraiserAddressBookGroupContacts &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&group_id=`*value*][`&group_ids=`*value*][`&auth=`*value*][`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

group_id

Optional. A unique ID of the group.

Type xsd:integer.

group_ids

Optional. A comma-separated list of groups IDs.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamraiserAddressBookGroupContactsResponse xsi:schemaLocation="http://
convio.com/crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd"
 xmlns="http://convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <addressBookContact>
    <id>1123</id>
    <firstName>firstName</firstName>
    <lastName>lastName</lastName>
    <email>example+1123@example.com</email>
  </addressBookContact>
</getTeamraiserAddressBookGroupContactsResponse>
```

JSON response

```
{"getTeamraiserAddressBookGroupContactsResponse":{"addressBookContact":
{"id":"1123","lastName":"lastName","email":"example
+1123@example.com","firstName":"firstName"}}}
```

getTeamraiserConfig

Provides additional configuration information about a TeamRaiser event.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraiserConfig &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*

Usage Notes

This method can be called anonymously, however, if you do not call this method within the context of a logged-in user, you must specify a valid Constituent ID as a cons_id parameter; otherwise the method will return an error.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamraiserConfigResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <teamraiserConfig>
    <acceptingDonations>true</acceptingDonations>
    <acceptingRegistrations>true</acceptingRegistrations>
    <familyRegistrationAllowed>true</familyRegistrationAllowed>
    <offlineGiftTypes>
      <cash>true</cash>
      <check>true</check>
      <credit>false</credit>
    </offlineGiftTypes>
    <companyAssociationAllowed>true</companyAssociationAllowed>
    <companyNewEntryAllowed>false</companyNewEntryAllowed>
    <participantCompanyAssociationAllowed>true</
participantCompanyAssociationAllowed>
    <participantCompanyNewEntryAllowed>false</
participantCompanyNewEntryAllowed>
    <offlineTeamGifts>MEMBERS</offlineTeamGifts>
    <personalBlogAllowed>false</personalBlogAllowed>
    <personalGoalAllowed>true</personalGoalAllowed>
    <personalOfflineGiftsAllowed>true</personalOfflineGiftsAllowed>
    <personalPageEditing>PARTICIPANTS</personalPageEditing>
    <personalPageLayoutAllowed>false</personalPageLayoutAllowed>
    <personalPageStatusAllowed>false</personalPageStatusAllowed>
    <personalPageStyleAllowed>false</personalPageStyleAllowed>
    <personalSurveyEditAllowed>false</personalSurveyEditAllowed>
    <personalPageSecondPhoto>false</personalPageSecondPhoto>
    <teamCaptainsMaximum>2</teamCaptainsMaximum>
    <teamSizeMaximum>15</teamSizeMaximum>
    <teamsAllowed>true</teamsAllowed>
    <fbConnectPublishFeedImageSrc>http://www.convio.com/image.gif</
fbConnectPublishFeedImageSrc>
    <fbConnectPublishFeedImageUrl>http://www.convio.com</
fbConnectPublishFeedImageUrl>
    <ociEnabled>true</ociEnabled>
    <ociBaseUrl>http://example.com/oci_base</ociBaseUrl>
    <categoryId>1</categoryId>
    <pagingMaxPageSize>500</pagingMaxPageSize>
    <defaultStationeryId>1001</defaultStationeryId>
    <standardRegistrationAllowed>true</standardRegistrationAllowed>
    <anonymousRegistrationAllowed>false</anonymousRegistrationAllowed>
    <screennameRegistrationAllowed>false</screennameRegistrationAllowed>
    <affiliateTypeId>1</affiliateTypeId>
    <leadFrId>1000</leadFrId>
    <eCommerceStoreId>1001</eCommerceStoreId>
    <eCommerceStoreName>Teamraiser Store</eCommerceStoreName>
```

```
    <eCommerceProductId>1005</eCommerceProductId>
    <eCommerceProductName>Teamraiser Product</eCommerceProductName>
    <additionalDonationAllowed>true</additionalDonationAllowed>
    <donationLevels>
      <manualAllowed>true</manualAllowed>
      <donationLevel>
        <order>1</order>
        <name>Gold</name>
        <amount>
          <decimal>50.0</decimal>
          <formatted>$50.00</formatted>
        </amount>
        <description>Gold donation level.</description>
      </donationLevel>
      <donationLevel>
        <order>2</order>
        <name>Silver</name>
        <amount>
          <decimal>25.0</decimal>
          <formatted>$25.00</formatted>
        </amount>
        <description>Silver Donation Level</description>
      </donationLevel>
    </donationLevels>
  </teamraiserConfig>
</getTeamraiserConfigResponse>
```

JSON response

```
{"getTeamraiserConfigResponse":{"teamraiserConfig":
{"standardRegistrationAllowed":"true","ociBaseUrl":"http://example.com/
oci_base","familyRegistrationAllowed":"true","personalSurveyEditAllowed":"false","pagingM
 Product","teamCaptainsMaximum":"2","screennameRegistrationAllowed":"false","personalPage
{"donationLevel":[{"amount":
{"formatted":"$50.00","decimal":"50.0"},"order":"1","description":"Gold
 donation level.","name":"Gold"},{"amount":
{"formatted":"$25.00","decimal":"25.0"},"order":"2","description":"Silver
 Donation
 Level","name":"Silver"}],"manualAllowed":"true"},"additionalDonationAllowed":"true","acc
www.convio.com/image.gif","fbConnectPublishFeedImageUrl":"http://
www.convio.com","categoryId":"1","eCommerceStoreId":"1001","defaultStationeryId":"1001","
{"check":"true","cash":"true","credit":"false"},"eCommerceStoreName":"Teamraiser
 Store","acceptingDonations":"true","personalPageStyleAllowed":"false"}}}
```

getTeamraiserEventManagerRoles

Retrieves the list of Event Manager Roles and their associated permissions

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	CRTeamraiserAPI	POST	Yes.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraiserEventManagerRoles&api_key=`*value*`&login_name=`*value*`&login_password=`*value*`&v=`*value*[`&center_id=`*value*][`&response_format=`*xml*|*json*][`&source=`*value*][`&sub_source=`*value*][`&suppress_response_codes=`*value*]

Usage Notes

This Server API method returns information about TeamRaiser Event Manager Roles defined in the system. Use it in conjunction with the *addTeamraiserEventManager* or *listTeamraiserEventManagers* methods to cross-reference a TeamRaiser Event Manager Role with its corresponding role_id.

API Administrator accounts that access this API must be assigned TeamRaiser API User permission under the TeamRaiser Management permission type.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamraiserEventManagerRolesResponse xsi:schemaLocation="http://convio.com/
crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>2</totalNumberResults>
  <eventManagerRole>
    <role_id>0</role_id>
    <role_name>Sample Event Manager Role 01</role_name>
    <permission>
      <permission_name>Sample Permission 01_01</permission_name>
      <allowed>true</allowed>
    </permission>
    <permission>
      <permission_name>Sample Permission 01_11</permission_name>
      <allowed>false</allowed>
    </permission>
    <permission>
      <permission_name>Sample Permission 01_21</permission_name>
      <allowed>true</allowed>
    </permission>
    <permission>
      <permission_name>Sample Permission 01_31</permission_name>
      <allowed>false</allowed>
    </permission>
    <permission>
      <permission_name>Sample Permission 01_41</permission_name>
      <allowed>true</allowed>
    </permission>
  </eventManagerRole>
  <eventManagerRole>
    <role_id>1</role_id>
    <role_name>Sample Event Manager Role 11</role_name>
    <permission>
      <permission_name>Sample Permission 11_01</permission_name>
      <allowed>true</allowed>
    </permission>
    <permission>
      <permission_name>Sample Permission 11_11</permission_name>
      <allowed>false</allowed>
    </permission>
    <permission>
      <permission_name>Sample Permission 11_21</permission_name>
      <allowed>true</allowed>
    </permission>
    <permission>
```

```
      <permission_name>Sample Permission 11_31</permission_name>
      <allowed>false</allowed>
    </permission>
    <permission>
      <permission_name>Sample Permission 11_41</permission_name>
      <allowed>true</allowed>
    </permission>
  </eventManagerRole>
</getTeamraiserEventManagerRolesResponse>
```

JSON response

```
{"getTeamraiserEventManagerRolesResponse":
{"totalNumberResults":"2","eventManagerRole":[{"role_id":"0","permission":
[{"allowed":"true","permission_name":"Sample Permission
 01_01"},{"allowed":"false","permission_name":"Sample Permission
 01_11"},{"allowed":"true","permission_name":"Sample Permission
 01_21"},{"allowed":"false","permission_name":"Sample Permission
 01_31"},{"allowed":"true","permission_name":"Sample Permission
 01_41"}],"role_name":"Sample Event Manager Role 01"},
{"role_id":"1","permission":[{"allowed":"true","permission_name":"Sample
 Permission 11_01"},{"allowed":"false","permission_name":"Sample
 Permission 11_11"},{"allowed":"true","permission_name":"Sample
 Permission 11_21"},{"allowed":"false","permission_name":"Sample
 Permission 11_31"},{"allowed":"true","permission_name":"Sample Permission
 11_41"}],"role_name":"Sample Event Manager Role 11"}]}}
```

getTeamraiserResultsData

Returns the results from a teamraiser create or update API call.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	SRTeamraiserAPI	GET, POST	Server.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraiserResultsData &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&job_id=`*value*

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRTeamraiserAPI?method=getTeamraiserResultsData & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& job_id =`*value*

Usage Notes

Given the Job ID returned by either of the TeamRaiser Data methods: *addTeamraiserData* or *updateTeamraiserData* , this method returns the status of the batch operation if it is still processing, or the results if it is complete.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

job_id

Required. A job ID returned by the addTeamraiserData or updateTeamraiserData method

Type xsd:integer.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

job_id

Required. A job ID returned by the addTeamraiserData or updateTeamraiserData method

Type xsd:integer.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamraiserCreateUpdateResponse jobId="123245456"
 xsi:schemaLocation="http://convio.com/crm/v1.0 http://service.convio.net/
xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/v1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <jobResult Id="0">
    <success>false</success>
    <eventTitle>Test Event</eventTitle>
    <errors>
      <createUpdateError>
        <error>Warning: tried to update locked field (LocalOfficeAttn)</error>
        <errorLevel>Warning</errorLevel>
      </createUpdateError>
      <createUpdateError>
        <error>Interest Category not found</error>
        <errorLevel>Error</errorLevel>
      </createUpdateError>
    </errors>
  </jobResult>
  <jobResult Id="1">
    <success>true</success>
    <eventTitle>Test Event 2</eventTitle>
    <frId>1000</frId>
  </jobResult>
</getTeamraiserCreateUpdateResponse>
```

JSON response

```
{"getTeamraiserCreateUpdateResponse":{"jobId":"123245456","jobResult":
[{"errors":{"createUpdateError":[{"error":"Warning: tried to update locked
 field (LocalOfficeAttn)","errorLevel":"Warning"},{"error":"Interest
 Category not found","errorLevel":"Error"}]},"eventTitle":"Test
 Event","Id":"0","success":"false"},{"frId":"1000","eventTitle":"Test Event
 2","Id":"1","success":"true"}]}}
```

getTeamraiserSuggestion

Provides a suggestion for what a participant should do next.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraiserSuggestion &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*

Usage Notes

This method can be called anonymously, however, if you do not call this method within the context of a logged-in user, you must specify a valid Constituent ID as a cons_id parameter; otherwise the method will return an error.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamraiserSuggestionResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <suggestion>
    <type>CONTACTS</type>
    <message>Add or import your contacts.</message>
  </suggestion>
</getTeamraiserSuggestionResponse>
```

JSON response

```
{"getTeamraiserSuggestionResponse":{"suggestion":{"message":"Add or import
 your contacts.","type":"CONTACTS"}}}
```

getTeamraisersByDistance

Returns a list of TeamRaiser events that match the search criteria provided. Search criteria includes the starting zip/postal code, the distance to search from that starting zip/postal code, and the units of distance to use (miles or kilometers).

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraisersByDistance &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&starting_postal=`*value* [`&distance_units=`*km* | *mi*] [`&event_type=`*value*] [`&full_search=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*]

[`&list_page_offset=`*value*][`&list_page_size=`*value*][`&list_record_offset=`*value*][`&list_sort_column=`*value*][`&public_event_type=`*value*][`&search_distance=`*value*]

Usage Notes

This **client-only** method returns a list of active TeamRaiser Events located based off distance from a starting zip or postal code. Use this to provide your users with the ability to locate the closest event to them.

Note: An Active event is an event that is Accepting Gifts, Accepting Registrations, or Accepting Gifts and Registrations.

Note: Currently, the distance search only supports Zip Codes within the United States.

This method shares several features with *getTeamraisersByInfo*, including the ability to limit the search to Teamraiser Events within a given Security Category or Primary Event Type. To limit the search to a given Security Category, use the list_category_id to provide the Category ID to limit the search to. To limit the search to a given Primary Event Type, use the event_type parameter to specify the name of the Primary Event Type. If you provide the name of a Primary Event Type that does not exist, an error message will be thrown.

You can use these parameters to limit the search in ways that your users may not be familiar with. One example would be that if your organization has several chapters, divisions or affiliates and each wants their own search page to locate only their events. You can include the security category ID for that affiliate as a hidden form field to ensure that the search returns only events within that affiliate's category. If you have separate programs within each affiliate and want each program to have it's own search page, include the primary event type that identifies that program and the security category id for that affiliate.

Additionally, the list that is returned supports several parameters allowing finer control of the results, including filtering on any of the returned elements, pagination, and sorting.

Paginating the List takes two parameters: list_page_size, which specifies the number of results to be retrieved for that page. You must also provide **one** of the following two parameters:

- list_page_offset: specifies which page of the results to retrieve. The first page of results has a list_page_offset of 0, page 2 has a list_page_offset of 1, and so on.
- list_record_offset: specifies which record the page should start with. This also uses numbering from 0, so the first record has a list_record_offset of 0, the 2nd record has a list_record_offset of 1, and so on.

Filtering the List takes two parameters:

- list_filter_column: name of the element to be filtered on.
- list_filter_text: text that the element specified by list_filter_column must contain.

Note: list_filter_text performs an exact match, and is case sensitive.

Sorting the List takes two parameters:

- list_sort_column: name of the element to be sorted by.
- list_ascending: Boolean value indicating whether the sort should be in ascending order

Note: You can paginate, filter, and sort the list all in one call.

Things to know when using this method:

- US Zip codes should be the 5-digit zip code.
- Searches may be performed in either Miles or Kilometers, but the maximum search distance is 200 (either 200 miles or 200 kilometers)
- Event Type limitations only support Primary Event Types.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

starting_postal

Required. The starting zip code to use for a distance search. For US Zip Codes, use the 5-digit zip code. Currently only supports US Zip Codes.

Type xsd:string.

distance_units

Optional. The Unit of distance measurements to use for a distance search. Can be either 'mi' for miles or 'km' for kilometers. Default value is 'mi'.

Type xsd:string.

Default is mi.

Options are:

- km
- mi

event_type

Optional. The name of the Primary Event Type to limit search results to. Omit this parameter to search on all Primary Event Types. Defaults to null.

Type xsd:string.

full_search

Optional. If TRUE, returns all events regardless of event status, otherwise returns only published events. Possible status values include: Unpublished = 0, Accepting Registrations Only = 1, AcceptingRegistrations and Gifts = 2, Accepting Gifts Only= 3, Closed = 4, Archived = 8, Deleted = 9.

Type xsd:boolean.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

public_event_type

Optional. The case-sensitive name of the Public Event Type to limit search results to. Omit this parameter to search on events with any Public Event Type. Defaults to null.

Type xsd:string.

search_distance

Optional. The radius to search within from the starting zip / postal code. Must be a number from 0-200, defaults to 0.

Type xsd:integer.

Default is 0.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamraisersResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>3</totalNumberResults>
  <teamraiser>
    <id>12000</id>
    <name>Sample Event Title 0</name>
    <description>Sample Event Description 0</description>
    <event_date>2013-02-03T05:18:26.789-06:00</event_date>
    <accepting_registrations>true</accepting_registrations>
    <accepting_donations>true</accepting_donations>
    <location_name>Sample Location 0</location_name>
    <street_address>1234 Fake St.</street_address>
    <city>Austin</city>
    <state>TX</state>
    <zip>78758</zip>
    <county>Travis</county>
    <country>United States</country>
    <region>Hill Country</region>
    <area>North Austin</area>
    <mailingAddrAttn>J. Doe</mailingAddrAttn>
    <mail_street1>1234 Fake Ave.</mail_street1>
    <mail_street2>Suite 1000</mail_street2>
    <mail_city>Austin</mail_city>
    <mail_state>TX</mail_state>
    <mail_zip>78758</mail_zip>
    <status>1</status>
    <greeting_url>http://example.com/greetingUrl</greeting_url>
    <find_url>http://example.com/findTeamOrParticipantUrl</find_url>
    <donate_event_url>http://example.com/donateUrl</donate_event_url>
```

```
    <reg_indiv_url>http://example.com/registerIndividualUrl</reg_indiv_url>
    <reg_new_team_url>http://example.com/formANewTeamUrl</reg_new_team_url>
    <reg_join_team_url>http://example.com/joinTeamUrl</reg_join_team_url>
    <public_event_type_name>Central Texas</public_event_type_name>
    <ecommerce_event_url>http://example.com/eCommerceEventUrl</
ecommerce_event_url>
    <distance>10.0</distance>
  </teamraiser>
  <teamraiser>
    <id>12001</id>
    <name>Sample Event Title 1</name>
    <description>Sample Event Description 1</description>
    <event_date>2013-02-03T05:18:26.792-06:00</event_date>
    <accepting_registrations>true</accepting_registrations>
    <accepting_donations>true</accepting_donations>
    <location_name>Sample Location 1</location_name>
    <street_address>1234 Fake St.</street_address>
    <city>Austin</city>
    <state>TX</state>
    <zip>78758</zip>
    <county>Travis</county>
    <country>United States</country>
    <region>Hill Country</region>
    <area>North Austin</area>
    <mailingAddrAttn>J. Doe</mailingAddrAttn>
    <mail_street1>1234 Fake Ave.</mail_street1>
    <mail_street2>Suite 1000</mail_street2>
    <mail_city>Austin</mail_city>
    <mail_state>TX</mail_state>
    <mail_zip>78758</mail_zip>
    <status>1</status>
    <greeting_url>http://example.com/greetingUrl</greeting_url>
    <find_url>http://example.com/findTeamOrParticipantUrl</find_url>
    <donate_event_url>http://example.com/donateUrl</donate_event_url>
    <reg_indiv_url>http://example.com/registerIndividualUrl</reg_indiv_url>
    <reg_new_team_url>http://example.com/formANewTeamUrl</reg_new_team_url>
    <reg_join_team_url>http://example.com/joinTeamUrl</reg_join_team_url>
    <public_event_type_name>Central Texas</public_event_type_name>
    <ecommerce_event_url>http://example.com/eCommerceEventUrl</
ecommerce_event_url>
    <distance>11.0</distance>
  </teamraiser>
  <teamraiser>
    <id>12002</id>
    <name>Sample Event Title 2</name>
    <description>Sample Event Description 2</description>
    <event_date>2013-02-03T05:18:26.793-06:00</event_date>
    <accepting_registrations>true</accepting_registrations>
    <accepting_donations>true</accepting_donations>
    <location_name>Sample Location 2</location_name>
    <street_address>1234 Fake St.</street_address>
    <city>Austin</city>
    <state>TX</state>
    <zip>78758</zip>
    <county>Travis</county>
    <country>United States</country>
    <region>Hill Country</region>
    <area>North Austin</area>
    <mailingAddrAttn>J. Doe</mailingAddrAttn>
    <mail_street1>1234 Fake Ave.</mail_street1>
    <mail_street2>Suite 1000</mail_street2>
    <mail_city>Austin</mail_city>
    <mail_state>TX</mail_state>
    <mail_zip>78758</mail_zip>
```

```
    <status>1</status>
    <greeting_url>http://example.com/greetingUrl</greeting_url>
    <find_url>http://example.com/findTeamOrParticipantUrl</find_url>
    <donate_event_url>http://example.com/donateUrl</donate_event_url>
    <reg_indiv_url>http://example.com/registerIndividualUrl</reg_indiv_url>
    <reg_new_team_url>http://example.com/formANewTeamUrl</reg_new_team_url>
    <reg_join_team_url>http://example.com/joinTeamUrl</reg_join_team_url>
    <public_event_type_name>Central Texas</public_event_type_name>
    <ecommerce_event_url>http://example.com/eCommerceEventUrl</
ecommerce_event_url>
    <distance>12.0</distance>
  </teamraiser>
</getTeamraisersResponse>
```

JSON response


```
{"getTeamraisersResponse":{"teamraiser":[{"region":"Hill
 Country","state":"TX","city":"Austin","id":"12000","distance":"10.0","reg_new_team_url":
example.com/formANewTeamUrl","mail_state":"TX","area":"North
 Austin","description":"Sample Event Description 0","name":"Sample
 Event Title 0","mail_city":"Austin","reg_indiv_url":"http://
example.com/registerIndividualUrl","ecommerce_event_url":"http://
example.com/eCommerceEventUrl","public_event_type_name":"Central
 Texas","zip":"78758","reg_join_team_url":"http://
example.com/joinTeamUrl","location_name":"Sample
 Location 0","status":"1","find_url":"http://example.com/
findTeamOrParticipantUrl","country":"United
 States","accepting_donations":"true","greeting_url":"http://
example.com/greetingUrl","donate_event_url":"http://example.com/
donateUrl","county":"Travis","event_date":"2013-02-03T05:18:26.796-06:00","mail_zip":"787
 Doe","street_address":"1234 Fake
 St.","accepting_registrations":"true","mail_street1":"1234
 Fake Ave.","mail_street2":"Suite 1000"},{"region":"Hill
 Country","state":"TX","city":"Austin","id":"12001","distance":"11.0","reg_new_team_url":
example.com/formANewTeamUrl","mail_state":"TX","area":"North
 Austin","description":"Sample Event Description 1","name":"Sample
 Event Title 1","mail_city":"Austin","reg_indiv_url":"http://
example.com/registerIndividualUrl","ecommerce_event_url":"http://
example.com/eCommerceEventUrl","public_event_type_name":"Central
 Texas","zip":"78758","reg_join_team_url":"http://
example.com/joinTeamUrl","location_name":"Sample
 Location 1","status":"1","find_url":"http://example.com/
findTeamOrParticipantUrl","country":"United
 States","accepting_donations":"true","greeting_url":"http://
example.com/greetingUrl","donate_event_url":"http://example.com/
donateUrl","county":"Travis","event_date":"2013-02-03T05:18:26.796-06:00","mail_zip":"787
 Doe","street_address":"1234 Fake
 St.","accepting_registrations":"true","mail_street1":"1234
 Fake Ave.","mail_street2":"Suite 1000"},{"region":"Hill
 Country","state":"TX","city":"Austin","id":"12002","distance":"12.0","reg_new_team_url":
example.com/formANewTeamUrl","mail_state":"TX","area":"North
 Austin","description":"Sample Event Description 2","name":"Sample
 Event Title 2","mail_city":"Austin","reg_indiv_url":"http://
example.com/registerIndividualUrl","ecommerce_event_url":"http://
example.com/eCommerceEventUrl","public_event_type_name":"Central
 Texas","zip":"78758","reg_join_team_url":"http://
example.com/joinTeamUrl","location_name":"Sample
 Location 2","status":"1","find_url":"http://example.com/
findTeamOrParticipantUrl","country":"United
 States","accepting_donations":"true","greeting_url":"http://
example.com/greetingUrl","donate_event_url":"http://example.com/
```

```
donateUrl","county":"Travis","event_date":"2013-02-03T05:18:26.797-06:00","mail_zip":"787
 Doe","street_address":"1234 Fake
 St.","accepting_registrations":"true","mail_street1":"1234 Fake
 Ave.","mail_street2":"Suite 1000"}],"totalNumberResults":"3"}}
```

Examples

To perform a Distance search looking for any events within 25 miles of the Zip Code 78758:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByDistance&starting_postal=78758&search_dista
```

Same as above, but limiting the search to events with a Primary Event Type of "Walks", and only retrieving the first page of results, displaying 10 results per page:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByDistance&starting_postal=78758&search_dista
```

Same as above, but sorting the results by distance:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByDistance&starting_postal=78758&search_dista
```

Retrieving Page 2 of the above query:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByDistance&starting_postal=78758&search_dista
```

A Distance search looking for any events within 200 kilometers of the Zip Code 20037:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByDistance&starting_postal=20037&search_dista
```

Same as above, but selecting 20 results per page starting with result #5, sorting by name in descending order, filtered to events with an area of Capitol Hill:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByDistance&starting_postal=20037&search_dista
%20Hill&list_sort_column=name&list_ascending=false
```

getTeamraisersByInfo

Returns a list of TeamRaiser events that match the search criteria provided. Search criteria includes the TeamRaiser's Public Event Name; the Event Information fields of: City, County, and State; Event Security Category; and Primary Event Type.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTeamraisersByInfo` `&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&name=`*value* [`&city=`*value*] [`&country=`*value*] [`&county=`*value*] [`&event_type=`*value*] [`&full_search=`*value*] [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&public_event_type=`*value*] [`&state=`*value*]

Usage Notes

This **client-only** method returns a list of active TeamRaiser Events located based off criteria that matches the Event's Public Name and/or the Event's Location Information.

Note: An Active event is an event that is Accepting Gifts, Accepting Registrations, or Accepting Gifts and Registrations.

Use this method when you want to provide your users the ability to locate events that they may already be familiar with by name, or to find events in the geographical area of their choice.

Note: When searching on the event information, the search is a like search, and is case insensitive. A search for `name=walk` will return all events that have the word "walk" in their name, such as the events "Walk Against Hunger" or "Survivors walking out".

When using this method, you must provide a value for at least 1 of the following parameters: name, city, county, or state. Each paramter is individually optional, but at least one of them must provide some data for the system to search on. If you do not provide a value to search on for one of those parameters, you will receive an error.

This method shares several features with *getTeamraisersByDistance*, including the ability to limit the search to TeamRaiser Events within a given Security Category or Primary Event Type. To limit the search to a given Security Category, use the list_category_id to provide the Category ID to limit the search to. To limit the search to a given Primary Event Type, use the event_type parameter to specify the name of the Primary Event Type. If you provide the name of a Primary Event Type that does not exist, an error message will be thrown.

You can use these parameters to limit the search in ways that your users may not be familiar with. One example would be that if your organization has several chapters, divisions or affiliates and each wants their own search page to locate only their events. You can include the security category ID for that affiliate as a hidden form field to ensure that the search returns only events within that affiliate's category. If you have separate programs within each affiliate and want each program to have it's own search page, include the primary event type that identifies that program and the security category id for that affiliate.

Additionally, the list that is returned supports several parameters allowing finer control of the results, including filtering on any of the returned elements, pagination, and sorting.

Paginating the List takes two parameters: list_page_size, which specifies the number of results to be retrieved for that page. You must also provide **one** of the following two parameters:

- list_page_offset: specifies which page of the results to retrieve. The first page of results has a list_page_offset of 0, page 2 has a list_page_offset of 1, and so on.
- list_record_offset: specifies which record the page should start with. This also uses numbering from 0, so the first record has a list_record_offset of 0, the 2nd record has a list_record_offset of 1, and so on.

Filtering the List takes two parameters:

- list_filter_column: name of the element to be filtered on.
- list_filter_text: text that the element specified by list_filter_column must contain.

Note: list_filter_text performs an exact match, and is case sensitive.

Sorting the List takes two parameters:

- list_sort_column: name of the element to be sorted by.
- list_ascending: Boolean value indicating whether the sort should be in ascending order

Note: You can paginate, filter, and sort the list all in one call.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

name

Required. The name of the TeamRaiser event.

Type xsd:string.

city

Optional. The city the event takes place in, as configured in the Event's Location Details.

Type xsd:string.

country

Optional. The country the event takes place in, as configured in the Event's Location Details.

Type xsd:string.

county

Optional. The county the event takes place in, as configured in the Event's Location Details.

Type xsd:string.

event_type

Optional. The name of the Primary Event Type to limit search results to. Omit this parameter to search on all Primary Event Types. Defaults to null.

Type xsd:string.

full_search

Optional. If TRUE, returns all events regardless of event status, otherwise returns only published events. Possible status values include: Unpublished = 0, Accepting Registrations Only = 1, AcceptingRegistrations and Gifts = 2, Accepting Gifts Only= 3, Closed = 4, Archived = 8, Deleted = 9.

Type xsd:boolean.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

public_event_type

Optional. The case-sensitive name of the Public Event Type to limit search results to. Omit this parameter to search on events with any Public Event Type. Defaults to null.

Type xsd:string.

state

Optional. The state the event takes place in, as configured in the Event's Location Details.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamraisersResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>3</totalNumberResults>
  <teamraiser>
    <id>12000</id>
    <name>Sample Event Title 0</name>
    <description>Sample Event Description 0</description>
    <event_date>2013-02-03T05:18:25.456-06:00</event_date>
    <accepting_registrations>true</accepting_registrations>
    <accepting_donations>true</accepting_donations>
    <location_name>Sample Location 0</location_name>
    <street_address>1234 Fake St.</street_address>
    <city>Austin</city>
    <state>TX</state>
    <zip>78758</zip>
    <county>Travis</county>
    <country>United States</country>
    <region>Hill Country</region>
    <area>North Austin</area>
    <mailingAddrAttn>J. Doe</mailingAddrAttn>
    <mail_street1>1234 Fake Ave.</mail_street1>
    <mail_street2>Suite 1000</mail_street2>
    <mail_city>Austin</mail_city>
    <mail_state>TX</mail_state>
    <mail_zip>78758</mail_zip>
    <status>1</status>
    <greeting_url>http://example.com/greetingUrl</greeting_url>
    <find_url>http://example.com/findTeamOrParticipantUrl</find_url>
    <donate_event_url>http://example.com/donateUrl</donate_event_url>
    <reg_indiv_url>http://example.com/registerIndividualUrl</reg_indiv_url>
    <reg_new_team_url>http://example.com/formANewTeamUrl</reg_new_team_url>
    <reg_join_team_url>http://example.com/joinTeamUrl</reg_join_team_url>
    <public_event_type_name>Central Texas</public_event_type_name>
```

```
    <ecommerce_event_url>http://example.com/eCommerceEventUrl</
ecommerce_event_url>
  </teamraiser>
  <teamraiser>
    <id>12001</id>
    <name>Sample Event Title 1</name>
    <description>Sample Event Description 1</description>
    <event_date>2013-02-03T05:18:25.456-06:00</event_date>
    <accepting_registrations>true</accepting_registrations>
    <accepting_donations>true</accepting_donations>
    <location_name>Sample Location 1</location_name>
    <street_address>1234 Fake St.</street_address>
    <city>Austin</city>
    <state>TX</state>
    <zip>78758</zip>
    <county>Travis</county>
    <country>United States</country>
    <region>Hill Country</region>
    <area>North Austin</area>
    <mailingAddrAttn>J. Doe</mailingAddrAttn>
    <mail_street1>1234 Fake Ave.</mail_street1>
    <mail_street2>Suite 1000</mail_street2>
    <mail_city>Austin</mail_city>
    <mail_state>TX</mail_state>
    <mail_zip>78758</mail_zip>
    <status>1</status>
    <greeting_url>http://example.com/greetingUrl</greeting_url>
    <find_url>http://example.com/findTeamOrParticipantUrl</find_url>
    <donate_event_url>http://example.com/donateUrl</donate_event_url>
    <reg_indiv_url>http://example.com/registerIndividualUrl</reg_indiv_url>
    <reg_new_team_url>http://example.com/formANewTeamUrl</reg_new_team_url>
    <reg_join_team_url>http://example.com/joinTeamUrl</reg_join_team_url>
    <public_event_type_name>Central Texas</public_event_type_name>
    <ecommerce_event_url>http://example.com/eCommerceEventUrl</
ecommerce_event_url>
  </teamraiser>
  <teamraiser>
    <id>12002</id>
    <name>Sample Event Title 2</name>
    <description>Sample Event Description 2</description>
    <event_date>2013-02-03T05:18:25.456-06:00</event_date>
    <accepting_registrations>true</accepting_registrations>
    <accepting_donations>true</accepting_donations>
    <location_name>Sample Location 2</location_name>
    <street_address>1234 Fake St.</street_address>
    <city>Austin</city>
    <state>TX</state>
    <zip>78758</zip>
    <county>Travis</county>
    <country>United States</country>
    <region>Hill Country</region>
    <area>North Austin</area>
    <mailingAddrAttn>J. Doe</mailingAddrAttn>
    <mail_street1>1234 Fake Ave.</mail_street1>
    <mail_street2>Suite 1000</mail_street2>
    <mail_city>Austin</mail_city>
    <mail_state>TX</mail_state>
    <mail_zip>78758</mail_zip>
    <status>1</status>
    <greeting_url>http://example.com/greetingUrl</greeting_url>
    <find_url>http://example.com/findTeamOrParticipantUrl</find_url>
    <donate_event_url>http://example.com/donateUrl</donate_event_url>
    <reg_indiv_url>http://example.com/registerIndividualUrl</reg_indiv_url>
    <reg_new_team_url>http://example.com/formANewTeamUrl</reg_new_team_url>
```

```
    <reg_join_team_url>http://example.com/joinTeamUrl</reg_join_team_url>
    <public_event_type_name>Central Texas</public_event_type_name>
    <ecommerce_event_url>http://example.com/eCommerceEventUrl</
ecommerce_event_url>
  </teamraiser>
</getTeamraisersResponse>
```

JSON response


```
{"getTeamraisersResponse":{"teamraiser":[{"region":"Hill
 Country","state":"TX","city":"Austin","id":"12000","reg_new_team_url":"http://
example.com/formANewTeamUrl","mail_state":"TX","area":"North
 Austin","description":"Sample Event Description 0","name":"Sample
 Event Title 0","mail_city":"Austin","reg_indiv_url":"http://
example.com/registerIndividualUrl","ecommerce_event_url":"http://
example.com/eCommerceEventUrl","public_event_type_name":"Central
 Texas","zip":"78758","reg_join_team_url":"http://
example.com/joinTeamUrl","location_name":"Sample
 Location 0","status":"1","find_url":"http://example.com/
findTeamOrParticipantUrl","country":"United
 States","accepting_donations":"true","greeting_url":"http://
example.com/greetingUrl","donate_event_url":"http://example.com/
donateUrl","county":"Travis","event_date":"2013-02-03T05:18:25.458-06:00","mail_zip":"787
 Doe","street_address":"1234 Fake
 St.","accepting_registrations":"true","mail_street1":"1234
 Fake Ave.","mail_street2":"Suite 1000"},{"region":"Hill
 Country","state":"TX","city":"Austin","id":"12001","reg_new_team_url":"http://
example.com/formANewTeamUrl","mail_state":"TX","area":"North
 Austin","description":"Sample Event Description 1","name":"Sample
 Event Title 1","mail_city":"Austin","reg_indiv_url":"http://
example.com/registerIndividualUrl","ecommerce_event_url":"http://
example.com/eCommerceEventUrl","public_event_type_name":"Central
 Texas","zip":"78758","reg_join_team_url":"http://
example.com/joinTeamUrl","location_name":"Sample
 Location 1","status":"1","find_url":"http://example.com/
findTeamOrParticipantUrl","country":"United
 States","accepting_donations":"true","greeting_url":"http://
example.com/greetingUrl","donate_event_url":"http://example.com/
donateUrl","county":"Travis","event_date":"2013-02-03T05:18:25.458-06:00","mail_zip":"787
 Doe","street_address":"1234 Fake
 St.","accepting_registrations":"true","mail_street1":"1234
 Fake Ave.","mail_street2":"Suite 1000"},{"region":"Hill
 Country","state":"TX","city":"Austin","id":"12002","reg_new_team_url":"http://
example.com/formANewTeamUrl","mail_state":"TX","area":"North
 Austin","description":"Sample Event Description 2","name":"Sample
 Event Title 2","mail_city":"Austin","reg_indiv_url":"http://
example.com/registerIndividualUrl","ecommerce_event_url":"http://
example.com/eCommerceEventUrl","public_event_type_name":"Central
 Texas","zip":"78758","reg_join_team_url":"http://
example.com/joinTeamUrl","location_name":"Sample
 Location 2","status":"1","find_url":"http://example.com/
findTeamOrParticipantUrl","country":"United
 States","accepting_donations":"true","greeting_url":"http://
example.com/greetingUrl","donate_event_url":"http://example.com/
donateUrl","county":"Travis","event_date":"2013-02-03T05:18:25.459-06:00","mail_zip":"787
 Doe","street_address":"1234 Fake
 St.","accepting_registrations":"true","mail_street1":"1234 Fake
 Ave.","mail_street2":"Suite 1000"}],"totalNumberResults":"3"}}
```

Examples

To perform an Information search looking for events with "fest" in their Name:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByInfo&name=fest
```

Same as above, but limiting the search to events with a Primary Event Type of "Walks", and only retrieving the first page of results, displaying 10 results per page:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByInfo&name=fest&event_type=Walks&list_page_s
```

Same as above, but sorting the results by event date:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByInfo&name=fest&event_type=Walks&list_page_s
```

Retrieving Page 2 of the above query:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByInfo&name=fest&event_type=Walks&list_page_s
```

An Event Information search for events in the state of New York, with the name of "Funwalk", in Oneida county, limited only to events within the Security Category with the id=1020:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByInfo&name=Funwalk&state=NY&county=oneida&li
```

Same as above, but selecting 20 results per page starting with result #5, sorting by name in descending order, filtered to events with an area of Capitol Hill:

```
https://demo-secure.convio.net/demo829/site/CRTeamraiserAPI?
api_key=MyAPIKey&v=1.0&method=getTeamraisersByInfo&name=Funwalk&state=NY&county=oneida&li
%20Hill&list_sort_column=name&list_ascending=false
```

getTentingSearch

Provides a list of tenting search results.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTentingSearch` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&list_ascending=`*value*] [`&list_category_id=`*value*] [`&list_filter_column=`*value*] [`&list_filter_text=`*value*] [`&list_page_offset=`*value*] [`&list_page_size=`*value*] [`&list_record_offset=`*value*] [`&list_sort_column=`*value*] [`&search_email=`*value*] [`&search_first_name=`*value*] [`&search_last_name=`*value*] [`&search_race_number=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

list_ascending

Optional. Whether the list should be sorted in ascending order.

Type xsd:boolean.

Default is true.

list_category_id

Optional. Security Category filter.

Type xsd:integer.

list_filter_column

Optional. The database column that will be used to match the filter text (used with "list_filter_text").

Type xsd:string.

list_filter_text

Optional. The text that will be used to filter results.

Type xsd:string.

list_page_offset

Optional. The requested page number (used with "list_page_size"). The first page (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_page_size

Optional. The number of results in each page (used with "list_record_offset" or "list_page_offset").

Type xsd:integer.

Default is 25.

list_record_offset

Optional. The requested initial record number (used with "list_page_size"). The first record (default) is zero. Only one of "list_record_offset" or "list_page_offset" should be used in a request.

Type xsd:integer.

list_sort_column

Optional. The column to sort with (used with "list_ascending").

Type xsd:string.

search_email

Optional. Email to search for.

Type xsd:string.

search_first_name

Optional. First name of participant to search for.

Type xsd:string.

search_last_name

Optional. Last name of participant to search for.

Type xsd:string.

search_race_number

Optional. Race number to search for.

Type xsd:integer.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTentingSearchResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <totalNumberResults>3</totalNumberResults>
  <record>
    <consId>1001000</consId>
    <eventId>1000</eventId>
    <tentmateStatus>0</tentmateStatus>
    <personalPageUrl>http://example.com/personal_page</personalPageUrl>
    <name>
      <first>Sam</first>
      <last>Ple0</last>
    </name>
  </record>
  <record>
    <consId>1001001</consId>
    <eventId>1001</eventId>
    <tentmateStatus>1</tentmateStatus>
    <personalPageUrl>http://example.com/personal_page</personalPageUrl>
    <name>
      <first>Sam</first>
      <last>Ple1</last>
    </name>
  </record>
  <record>
    <consId>1001002</consId>
    <eventId>1002</eventId>
    <tentmateStatus>2</tentmateStatus>
    <personalPageUrl>http://example.com/personal_page</personalPageUrl>
    <name>
      <first>Sam</first>
      <last>Ple2</last>
    </name>
  </record>
</getTentingSearchResponse>
```

JSON response

```
{"getTentingSearchResponse":{"record":
[{"tentmateStatus":"0","eventId":"1000","consId":"1001000","name":
{"last":"Ple0","first":"Sam"},"personalPageUrl":"http://example.com/
```

```
personal_page"},
{"tentmateStatus":"1","eventId":"1001","consId":"1001001","name":
{"last":"Ple1","first":"Sam"},"personalPageUrl":"http://example.com/
personal_page"},
{"tentmateStatus":"2","eventId":"1002","consId":"1001002","name":
{"last":"Ple2","first":"Sam"},"personalPageUrl":"http://example.com/
personal_page"}],"totalNumberResults":"3"}}
```

getTentmate

Provides your tentmate if you have one.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTentmate`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTentmateResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <record>
    <consId>1001001</consId>
    <eventId>1001</eventId>
```

```
    <tentmateStatus>1</tentmateStatus>
    <personalPageUrl>http://example.com/personal_page</personalPageUrl>
    <name>
      <first>Sam</first>
      <last>Ple</last>
    </name>
  </record>
</getTentmateResponse>
```

JSON response

```
{"getTentmateResponse":{"record":
{"tentmateStatus":"1","eventId":"1001","consId":"1001001","name":
{"last":"Ple","first":"Sam"},"personalPageUrl":"http://example.com/
personal_page"}}}
```

getTopCompaniesData

Provides a list of the top ten companies for a TeamRaiser event with corresponding values for the amount each has raised.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTopCompaniesData` `&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTopCompaniesDataResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <teamraiserData>
    <name>Sample Company 1</name>
    <total>$1200.00</total>
  </teamraiserData>
  <teamraiserData>
```

```
    <name>Sample Company 2</name>
    <total>$1190.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Company 3</name>
    <total>$1180.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Company 4</name>
    <total>$1170.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Company 5</name>
    <total>$1160.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Company 6</name>
    <total>$1150.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Company 7</name>
    <total>$1140.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Company 8</name>
    <total>$1130.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Company 9</name>
    <total>$1120.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Company 10</name>
    <total>$1110.00</total>
  </teamraiserData>
</getTopCompaniesDataResponse>
```

JSON response

```
{"getTopCompaniesDataResponse":{"teamraiserData":
[{"total":"$1200.00","name":"Sample Company 1"},
{"total":"$1190.00","name":"Sample Company 2"},
{"total":"$1180.00","name":"Sample Company 3"},
{"total":"$1170.00","name":"Sample Company 4"},
{"total":"$1160.00","name":"Sample Company 5"},
{"total":"$1150.00","name":"Sample Company 6"},
{"total":"$1140.00","name":"Sample Company 7"},
{"total":"$1130.00","name":"Sample Company 8"},
{"total":"$1120.00","name":"Sample Company 9"},
{"total":"$1110.00","name":"Sample Company 10"}]}}
```

getTopDonors

Provides a list of top donors for a participant with corresponding values for the amount each has raised.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTopDonors`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* [`&limit=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

limit

Optional. Limit

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTopDonorsResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <teamraiserData>
    <name>Sample Donor 01</name>
    <total>$100.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 11</name>
    <total>$50.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 21</name>
    <total>$33.33</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 31</name>
    <total>$25.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 41</name>
    <total>$20.00</total>
```

```
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 51</name>
    <total>$16.66</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 61</name>
    <total>$14.28</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 71</name>
    <total>$12.50</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 81</name>
    <total>$11.11</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 91</name>
    <total>$10.00</total>
  </teamraiserData>
</getTopDonorsResponse>
```

JSON response

```
{"getTopDonorsResponse":{"teamraiserData":[{"total":"$100.00","name":"Sample
 Donor 01"},{"total":"$50.00","name":"Sample Donor 11"},
{"total":"$33.33","name":"Sample Donor 21"},{"total":"$25.00","name":"Sample
 Donor 31"},{"total":"$20.00","name":"Sample Donor 41"},
{"total":"$16.66","name":"Sample Donor 51"},{"total":"$14.28","name":"Sample
 Donor 61"},{"total":"$12.50","name":"Sample Donor 71"},
{"total":"$11.11","name":"Sample Donor 81"},{"total":"$10.00","name":"Sample
 Donor 91"}]}}
```

getTopParticipantsData

Provides a list of the top ten participants for a TeamRaiser event with corresponding values for the amount each has raised.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTopParticipantsData &api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTopParticipantsDataResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <teamraiserData>
    <id>1001</id>
    <name>Sample Participant 1</name>
    <total>$1200.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1002</id>
    <name>Sample Participant 2</name>
    <total>$1190.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1003</id>
    <name>Sample Participant 3</name>
    <total>$1180.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1004</id>
    <name>Sample Participant 4</name>
    <total>$1170.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1005</id>
    <name>Sample Participant 5</name>
    <total>$1160.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1006</id>
    <name>Sample Participant 6</name>
    <total>$1150.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1007</id>
    <name>Sample Participant 7</name>
    <total>$1140.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1008</id>
    <name>Sample Participant 8</name>
    <total>$1130.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1009</id>
    <name>Sample Participant 9</name>
    <total>$1120.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1010</id>
    <name>Sample Participant 10</name>
    <total>$1110.00</total>
  </teamraiserData>
</getTopParticipantsDataResponse>
```

JSON response

```
{"getTopParticipantsDataResponse":{"teamraiserData":
[{"total":"$1200.00","id":"1001","name":"Sample Participant 1"},
{"total":"$1190.00","id":"1002","name":"Sample Participant 2"},
{"total":"$1180.00","id":"1003","name":"Sample Participant 3"},
{"total":"$1170.00","id":"1004","name":"Sample Participant 4"},
{"total":"$1160.00","id":"1005","name":"Sample Participant 5"},
{"total":"$1150.00","id":"1006","name":"Sample Participant 6"},
{"total":"$1140.00","id":"1007","name":"Sample Participant 7"},
{"total":"$1130.00","id":"1008","name":"Sample Participant 8"},
{"total":"$1120.00","id":"1009","name":"Sample Participant 9"},
{"total":"$1110.00","id":"1010","name":"Sample Participant 10"}]}}
```

getTopTeamDonors

Provides a list of top donors for a participant with corresponding values for the amount each has raised.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTopTeamDonors`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* [`&limit=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

limit

Optional. Limit

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTopTeamDonorsResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <teamraiserData>
    <name>Sample Donor 01</name>
    <total>$100.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 11</name>
    <total>$50.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 21</name>
    <total>$33.33</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 31</name>
    <total>$25.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 41</name>
    <total>$20.00</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 51</name>
    <total>$16.66</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 61</name>
    <total>$14.28</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 71</name>
    <total>$12.50</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 81</name>
    <total>$11.11</total>
  </teamraiserData>
  <teamraiserData>
    <name>Sample Donor 91</name>
    <total>$10.00</total>
  </teamraiserData>
</getTopTeamDonorsResponse>
```

JSON response

```
{"getTopTeamDonorsResponse":{"teamraiserData":
[{"total":"$100.00","name":"Sample Donor 01"},{"total":"$50.00","name":"Sample
 Donor 11"},{"total":"$33.33","name":"Sample Donor 21"},
{"total":"$25.00","name":"Sample Donor 31"},{"total":"$20.00","name":"Sample
 Donor 41"},{"total":"$16.66","name":"Sample Donor 51"},
{"total":"$14.28","name":"Sample Donor 61"},{"total":"$12.50","name":"Sample
 Donor 71"},{"total":"$11.11","name":"Sample Donor 81"},
{"total":"$10.00","name":"Sample Donor 91"}]}}
```

getTopTeamsData

Provides a list of the top ten teams for a TeamRaiser event with corresponding values for the amount each has raised.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getTopTeamsData`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [
`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*]
[`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTopTeamsDataResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <teamraiserData>
    <id>1001</id>
    <name>Sample Team 1</name>
    <total>$1200.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1002</id>
    <name>Sample Team 2</name>
    <total>$1190.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1003</id>
    <name>Sample Team 3</name>
    <total>$1180.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1004</id>
    <name>Sample Team 4</name>
    <total>$1170.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1005</id>
    <name>Sample Team 5</name>
    <total>$1160.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1006</id>
    <name>Sample Team 6</name>
```

```
    <total>$1150.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1007</id>
    <name>Sample Team 7</name>
    <total>$1140.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1008</id>
    <name>Sample Team 8</name>
    <total>$1130.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1009</id>
    <name>Sample Team 9</name>
    <total>$1120.00</total>
  </teamraiserData>
  <teamraiserData>
    <id>1010</id>
    <name>Sample Team 10</name>
    <total>$1110.00</total>
  </teamraiserData>
</getTopTeamsDataResponse>
```

JSON response

```
{"getTopTeamsDataResponse":{"teamraiserData":
[{"total":"$1200.00","id":"1001","name":"Sample Team 1"},
{"total":"$1190.00","id":"1002","name":"Sample Team 2"},
{"total":"$1180.00","id":"1003","name":"Sample Team 3"},
{"total":"$1170.00","id":"1004","name":"Sample Team 4"},
{"total":"$1160.00","id":"1005","name":"Sample Team 5"},
{"total":"$1150.00","id":"1006","name":"Sample Team 6"},
{"total":"$1140.00","id":"1007","name":"Sample Team 7"},
{"total":"$1130.00","id":"1008","name":"Sample Team 8"},
{"total":"$1120.00","id":"1009","name":"Sample Team 9"},
{"total":"$1110.00","id":"1010","name":"Sample Team 10"}]}}
```

getUpsell

Provides detailed information about an upsell.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=getUpsell`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][
`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*
][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&fr_id=`*value*
`&upsell_id=`*value*

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

upsell_id

Required. A unique ID for an upsell.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getUpsellResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <upsell>
    <name>Sample Upsell 1</name>
    <description>Upsell Description 1</description>
    <label>Upsell label</label>
    <id>101</id>
    <price>1001</price>
    <fairMarketValue>1001</fairMarketValue>
    <moreInfoLink>http://example.com/moreinfo</moreInfoLink>
    <required>false</required>
    <order>1</order>
    <option>
      <additionalCharge>0</additionalCharge>
      <text>Size 0</text>
      <id>201</id>
      <order>0</order>
      <display>true</display>
    </option>
    <option>
      <additionalCharge>0</additionalCharge>
      <text>Size 1</text>
      <id>202</id>
      <order>1</order>
      <display>true</display>
    </option>
    <option>
      <additionalCharge>0</additionalCharge>
      <text>Size 2</text>
      <id>203</id>
      <order>2</order>
      <display>true</display>
    </option>
  </upsell>
</getUpsellResponse>
```

JSON response

```
{"getUpsellResponse":{"upsell":
{"fairMarketValue":"1001","id":"101","price":"1001","order":"1","moreInfoLink":"http://
example.com/moreinfo","description":"Upsell Description 1","name":"Sample
 Upsell 1","label":"Upsell label","required":"false","option":
[{"id":"201","text":"Size
```

```
 0","order":"0","display":"true","additionalCharge":"0"},
{"id":"202","text":"Size
 1","order":"1","display":"true","additionalCharge":"0"},
{"id":"203","text":"Size
 2","order":"2","display":"true","additionalCharge":"0"}]}}}
```

joinTeam

The joinTeam API is used by a participant to join an existing TeamRaiser team for a given event.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=joinTeam`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*
`&team_id=`*value* [`&team_password=`*value*]

Usage Notes

The `joinTeam` method makes a participant who is registered for a TeamRaiser event as an individual a member of the specified team. To change a participant's team membership from one team to another, first call `leaveTeam` for the participant's current team, then `joinTeam` for the new team.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

team_id

Required. A unique team ID.

Type xsd:nonNegativeInteger.

team_password

Optional. Team password.

Type xsd:string.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<joinTeamResponse xmlns="http://convio.com/crm/v1.0" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
  <success xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd">true</success>
  <team>
    <name>Example Team</name>
```

```
    <description>Example team description</description>
    <captainsMessage>Example captains message</captainsMessage>
    <goal>1000</goal>
    <id>10000</id>
  </team>
</joinTeamResponse>
```

JSON response

```
{"joinTeamResponse":{"team":{"id":"10000","description":"Example team
 description","name":"Example Team","captainsMessage":"Example captains
 message","goal":"1000"},"success":
{"content":"true","xsi:schemaLocation":"http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd"}}}
```

leaveTeam

The leaveTeam method can be used to by a participant to leave a teamraiser team and participate as an indiviual.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=leaveTeam`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value*

Usage Notes

The `leaveTeam` method removes a participant from the specified TeamRaiser team so that they may participate in the TeamRaiser event as an individual or join a different team. A team captain can only leave a team if there are no other team members on that team. If the team captain leaves the team, the team is deleted.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<leaveTeamResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
</leaveTeamResponse>
```

JSON response

```
{"leaveTeamResponse":{"success":"true"}}
```

listTeamraiserEventManagers

Lists the Event Managers for the given TeamRaiser Event.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRTeamraiserAPI	POST	No.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRTeamraiserAPI?method=listTeamraiserEventManagers & api_key =`*value* `& login_name =`*value* `& login_password =`*value* `& v =`*value* [`& center_id =`*value*] [`& response_format =`*xml* | *json*] [`& source =`*value*] [`& sub_source =`*value*] [`& suppress_response_codes =`*value*] `& fr_id =`*value*

Usage Notes

This Server API method returns the list of TeamRaiser Event Managers assigned to a specified TeamRaiser event. You may use it in conjunction with *addTeamraiserEventManager*, *removeTeamraiserEventManager* and *getTeamraiserEventManagerRoles* methods to query, add or update Event Managers for TeamRaiser events.

API Administrator accounts that access this API must be assigned TeamRaiser API User permission under the TeamRaiser Management permission type.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<listTeamraiserEventManagersResponse fr_id="1000" xmlns="http://convio.com/
crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <event_manager xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd">
    <cons_id>1001001</cons_id>
    <role_id>101</role_id>
  </event_manager>
  <event_manager>
    <cons_id>1001002</cons_id>
    <role_id>102</role_id>
  </event_manager>
</listTeamraiserEventManagersResponse>
```

JSON response

```
{"listTeamraiserEventManagersResponse":{"fr_id":"1000","event_manager":
[{"role_id":"101","cons_id":"1001001","xsi:schemaLocation":"http://
convio.com/crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd"},
{"role_id":"102","cons_id":"1001002"}]}}
```

previewMessage

Returns a sample message.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=previewMessage &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&layout_id=`*value* `&subject=`*value* [`&message_body=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

layout_id

Required. The layout ID.

Type xsd:nonNegativeInteger.

subject

Required. The message subject.

Type xsd:string.

message_body

Optional. The message body.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getMessagePreviewResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <message>example html message here.</message>
</getMessagePreviewResponse>
```

JSON response

```
{"getMessagePreviewResponse":{"message":"example html message here."}}
```

processRegistration

Accepts an XML document describing a primary registrationDocument and zero or more secondary registrations and processes the registrations.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	SRTeamraiserAPI	POST	Server.

Client API Syntax
`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=processRegistration` `&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&registration_document=`*value* [`&match_by_username=`*value*] [`&suppress_registration_autoresponders=`*value*] [`&update_constituent_profile=`*value*]
Server API Syntax
`https://`*secure2.convio.net/organization*`/site/SRTeamraiserAPI?method=processRegistration` `&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] `&registration_document=`*value* [`&match_by_username=`*value*] [`&suppress_registration_autoresponders=`*value*] [`&update_constituent_profile=`*value*]

Usage Notes

Request Document

This method accepts an XML document with data for one primary TeamRaiser event registration and, if multiple registrations are allowed, one or more secondary registrations in the same event. The XML document follows the same format as the return document of *getRegistrationDocument*:

```
<processRegistrationRequest xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <eventId>1000</eventId>
 <primaryRegistration>
```

```
  <partTypeId>1100</partTypeId>
  <consId xsi:nil="true"/>
  <email required="true"/>
  <firstName required="true"/>
  <lastName required="true"/>
  <question>
   <id>1010</id>
   <type>PARTICIPATION_TYPE</type>
   <response xsi:nil="true"/>
   <key>T_SHIRT_SIZE</key>
   <hidden>false</hidden>
   <questionRequired>true</questionRequired>
  </question>
 </primaryRegistration>
 <secondaryRegistration>
  <partTypeId>1100</partTypeId>
  <consId xsi:nil="true"/>
  <email required="true"/>
  <firstName required="true"/>
  <lastName required="true"/>
  <question>
   <id>1010</id>
   <type>EVENT</type>
   <response xsi:nil="true"/>
   <key>Turned_In_Permission_Slip</key>
   <hidden>true</hidden>
   <questionRequired>true</questionRequired>
  </question>
 </secondaryRegistration>
</processRegistrationRequest>
```

See *getRegistrationDocument* for a complete example and more information about the request document.

Constituent Matching

When processing the primary registration the method will attempt to find an existing constituent that matches the data in the request document. In order, the matching steps are:

1. If `<consId>` is specified the constituent with that constituent ID will be used. If there is not a constituent with that ID an error will be returned.
2. If `<memberId>` is specified the constituent with that member ID will be used. If there is not a constituent with that ID an error will be returned.
3. If `<username>` is specified the constituent with that username will be used.
4. If `<email>`, `<firstName>` and `<lastName>` are specified the constituent with that email address, first name and last name will be used. If more than one constituent matches an error will be returned.
5. If a matching constituent is not found create a new constituent.

Autoresponder Suppression

If the parameter suppress_registration_autoresponders is set to *true* the following autoresponders will not be sent to primary or secondary registrants:

- Thanks for Creating an Account with Site
- Thanks for Registering
- Thanks for Forming a Team
- Sample Solicitation

Payment Information

When calling this method via the Client API, you may specify `CREDIT_CARD` in the `<paymentType>` element and pass credit card information for processing, or you can specify `CASH` or `CHECK` to record a cash or check transaction. If you are calling this method via the Server API, you may only record `CASH` or `CHECK` transactions; credit card payment processing is not allowed.

Note: If the selected event and participation type combination is not currently accepting registrations, this method will return an error. This method can return an error even after getRegistrationDocument method succeeds if another registration causes an event or participation type limit to be reached between the two calls.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

registration_document

Required. The XML document containing the data for the primary registration and any secondary registrations.

Type xsd:string.

Default is .

match_by_username

Optional. If set to true an existing constituent that matches the username provided in the registration document will be registered for the event.

Type xsd:boolean.

Default is false.

suppress_registration_autoresponders

Optional. Prevent autoresponders from being sent to the constituent(s) being registered.

Type xsd:boolean.

Default is false.

update_constituent_profile

Optional. If set to true and an existing constituent is matched for the primary registrant the constituent profile will be updated. Secondary registrant information is never used for updates. It is only used for creating new constituents.

Type xsd:boolean.

Default is false.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

registration_document

Required. The XML document containing the data for the primary registration and any secondary registrations.

Type xsd:string.

Default is .

match_by_username

Optional. If set to true an existing constituent that matches the username provided in the registration document will be registered for the event.

Type xsd:boolean.

Default is false.

suppress_registration_autoresponders

Optional. Prevent autoresponders from being sent to the constituent(s) being registered.

Type xsd:boolean.

Default is false.

update_constituent_profile

Optional. If set to true and an existing constituent is matched for the primary registrant the constituent profile will be updated. Secondary registrant information is never used for updates. It is only used for creating new constituents.

Type xsd:boolean.

Default is false.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<processRegistrationResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <transactionId>1325</transactionId>
  <confirmationCode>1-1001-20-1001-1001</confirmationCode>
  <primaryRegistration>
    <registrationId>1001</registrationId>
    <consId>1100000</consId>
  </primaryRegistration>
  <secondaryRegistration>
    <registrationId>1002</registrationId>
    <consId>1100001</consId>
  </secondaryRegistration>
  <secondaryRegistration>
    <registrationId>1003</registrationId>
    <consId>1100002</consId>
  </secondaryRegistration>
</processRegistrationResponse>
```

JSON response

```
{"processRegistrationResponse":{"secondaryRegistration":
[{"consId":"1100001","registrationId":"1002"},
{"consId":"1100002","registrationId":"1003"}],"transactionId":"1325","primaryRegistration
{"consId":"1100000","registrationId":"1001"},"confirmationCode":"1-1001-20-1001-1001"}}
```

removeCompanyPhoto

Removes a company page photo.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=removeCompanyPhoto &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&company_id=`*value* `&fr_id=`*value* `&photo_1_delete=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method requires authentication, and must include the company_id. Only the logged-in constituent who is identified as the Company Coordinator in the Company record has permission to execute this method.

The Company Coordinator is defined by the Convio system administrator when the Company record is created.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

company_id

Required. The unique ID of the company.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

photo_1_delete

Required. If TRUE, confirms that the company photo should be deleted.

Type xsd:boolean.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<removeCompanyPhotoResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <photoItem>
    <id>123</id>
    <thumbnailUrl>http://www.example.com/thumbnailexample.jpg</thumbnailUrl>
  </photoItem>
</removeCompanyPhotoResponse>
```

JSON response

```
{"removeCompanyPhotoResponse":{"photoItem":{"id":"123","thumbnailUrl":"http://
www.example.com/thumbnailexample.jpg"}}}
```

removePersonalPhoto

Removes a personal page photo.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=removePersonalPhoto &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&graphic_upload2_delete=`*value*] [`&graphic_upload_delete=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

graphic_upload2_delete

Optional. Indicates that photo 2 of the personal page should be deleted.

graphic_upload_delete

Optional. Indicates that photo 1 of the personal page should be deleted.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<removePersonalPhotoResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <photoItem>
```

```
    <id>123</id>
    <thumbnailUrl>http://www.example.com/thumbnailexample.jpg</thumbnailUrl>
  </photoItem>
</removePersonalPhotoResponse>
```

JSON response

```
{"removePersonalPhotoResponse":{"photoItem":
{"id":"123","thumbnailUrl":"http://www.example.com/thumbnailexample.jpg"}}}
```

removeTeamPhoto

Removes a team page photo.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=removeTeamPhoto` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&graphic_upload_delete=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method requires authentication. The logged-in user must be a Team Captain to execute this method successfully.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

graphic_upload_delete

Optional. Required parameter indicating the team photo should be deleted.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<removeTeamPhotoResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <photoItem>
    <id>123</id>
    <thumbnailUrl>http://www.example.com/thumbnailexample.jpg</thumbnailUrl>
  </photoItem>
</removeTeamPhotoResponse>
```

JSON response

```
{"removeTeamPhotoResponse":{"photoItem":{"id":"123","thumbnailUrl":"http://
www.example.com/thumbnailexample.jpg"}}}
```

removeTeamraiserEventManager

Removes a constituent from the list of Event Managers for a TeamRaiser Event.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRTeamraiserAPI	POST	No.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRTeamraiserAPI?method=removeTeamraiserEventManager&api_key=`*value*`&login_name=`*value*`&login_password=`*value*`&v=`*value*`[&center_id=`*value*`][&response_format=`*xml*`|`*json*`][&source=`*value*`][&sub_source=`*value*`][&suppress_response_codes=`*value*`]&cons_id=`*value*`&fr_id=`*value*

Usage Notes

Use this method to remove a specified constituent from the list of Event Managers for a given TeamRaiser Event.

This Server API is accessible only from white-listed servers. API Administrator accounts that access this API must be assigned TeamRaiser API User permission under the TeamRaiser Management permission type.

See also: *listTeamraiserEventManagers*, *addTeamraiserEventManager* and *getTeamraiserEventManagerRoles*.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

cons_id

Required. Constituent ID

Type xsd:nonNegativeInteger.

fr_id

Required. A unique Teamraiser ID

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<removeTeamraiserEventManagerResponse xmlns="http://convio.com/crm/v1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd">true</success>
</removeTeamraiserEventManagerResponse>
```

JSON response

```
{"removeTeamraiserEventManagerResponse":{"success":
{"content":"true","xsi:schemaLocation":"http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd"}}}
```

sendTafMessage

Send a message to a set of email addresses.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=sendTafMessage` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&layout_id=`*value* `&message_id=`*value* `&message_name=`*value* `&recipients=`*value* `&subject=`*value* [`&message_body=`*value*] [`&prepend_salutation=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

layout_id

Required. The layout ID.

Type xsd:nonNegativeInteger.

message_id

Required. Message ID

Type xsd:nonNegativeInteger.

message_name

Required. The name of the message.

Type xsd:string.

recipients

Required. The message recipients list.

Type xsd:string.

subject

Required. The message subject.

Type xsd:string.

message_body

Optional. The message body.

Type xsd:string.

prepend_salutation

Optional. Setting to prepend personalized salutation to TAF emails

Type xsd:boolean.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<sendTafMessageResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
</sendTafMessageResponse>
```

JSON response

```
{"sendTafMessageResponse":{"success":"true"}}
```

setTeamCaptains

Sets one or more Team Members as Team Captains.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	GET, POST	No.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=setTeamCaptains`
`&api_key=`*value* `&v=`*value* [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [
`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*

][&success_redirect=*value*][&suppress_response_codes=*value*] &captains=*value* &fr_id=*value* &team_id=*value*

Usage Notes

This method requires authentication. The logged-in user must be a Team Captain to execute this method successfully.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

captains

Required. Comma separated list of constituent IDs who should be captains of this team. Each constituent must already be a member of the team.

Type xsd:string.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

team_id

Required. A unique team ID.

Type xsd:nonNegativeInteger.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<getTeamMembersResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <member>
    <name>
      <first>Member</first>
      <last>Sample01</last>
    </name>
    <aTeamCaptain>true</aTeamCaptain>
    <teamName>Sample Team Name</teamName>
    <personalPageUrl>http://www.convio.net/site/TR/RunEvent/General?
px=1002000&amp;pg=personal&amp;fr_id=1000</personalPageUrl>
    <donationUrl>https://www.convio.net/org/site/Donation2?
df_id=1000&amp;FR_ID=1000&amp;PROXY_ID=1002000&amp;PROXY_TYPE=20</donationUrl>
    <teamPageUrl>http://www.convio.net/site/TR/RunEvent/General?
team_id=1000&amp;pg=team&amp;fr_id=1000</teamPageUrl>
    <consId>1002000</consId>
  </member>
  <member>
    <name>
      <first>Member</first>
      <last>Sample11</last>
    </name>
    <aTeamCaptain>true</aTeamCaptain>
    <teamName>Sample Team Name</teamName>
    <personalPageUrl>http://www.convio.net/site/TR/RunEvent/General?
px=1002001&amp;pg=personal&amp;fr_id=1000</personalPageUrl>
    <donationUrl>https://www.convio.net/org/site/Donation2?
df_id=1000&amp;FR_ID=1000&amp;PROXY_ID=1002001&amp;PROXY_TYPE=20</donationUrl>
```

```
    <teamPageUrl>http://www.convio.net/site/TR/RunEvent/General?
team_id=1000&amp;pg=team&amp;fr_id=1000</teamPageUrl>
    <consId>1002001</consId>
  </member>
  <member>
    <name>
      <first>Member</first>
      <last>Sample21</last>
    </name>
    <aTeamCaptain>false</aTeamCaptain>
    <teamName>Sample Team Name</teamName>
    <personalPageUrl>http://www.convio.net/site/TR/RunEvent/General?
px=1002002&amp;pg=personal&amp;fr_id=1000</personalPageUrl>
    <donationUrl>https://www.convio.net/org/site/Donation2?
df_id=1000&amp;FR_ID=1000&amp;PROXY_ID=1002002&amp;PROXY_TYPE=20</donationUrl>
    <teamPageUrl>http://www.convio.net/site/TR/RunEvent/General?
team_id=1000&amp;pg=team&amp;fr_id=1000</teamPageUrl>
    <consId>1002002</consId>
  </member>
</getTeamMembersResponse>
```

JSON response

```
{"getTeamMembersResponse":{"member":[{"teamName":"Sample Team
 Name","aTeamCaptain":"true","teamPageUrl":"http://www.convio.net/site/TR/
RunEvent/General?
team_id=1000&pg=team&fr_id=1000","consId":"1002000","donationUrl":"https://
www.convio.net/org/site/Donation2?
df_id=1000&FR_ID=1000&PROXY_ID=1002000&PROXY_TYPE=20","name":
{"last":"Sample01","first":"Member"},"personalPageUrl":"http://
www.convio.net/site/TR/RunEvent/General?px=1002000&pg=personal&fr_id=1000"},
{"teamName":"Sample Team Name","aTeamCaptain":"true","teamPageUrl":"http://
www.convio.net/site/TR/RunEvent/General?
team_id=1000&pg=team&fr_id=1000","consId":"1002001","donationUrl":"https://
www.convio.net/org/site/Donation2?
df_id=1000&FR_ID=1000&PROXY_ID=1002001&PROXY_TYPE=20","name":
{"last":"Sample11","first":"Member"},"personalPageUrl":"http://
www.convio.net/site/TR/RunEvent/General?px=1002001&pg=personal&fr_id=1000"},
{"teamName":"Sample Team Name","aTeamCaptain":"false","teamPageUrl":"http://
www.convio.net/site/TR/RunEvent/General?
team_id=1000&pg=team&fr_id=1000","consId":"1002002","donationUrl":"https://
www.convio.net/org/site/Donation2?
df_id=1000&FR_ID=1000&PROXY_ID=1002002&PROXY_TYPE=20","name":
{"last":"Sample21","first":"Member"},"personalPageUrl":"http://www.convio.net/
site/TR/RunEvent/General?px=1002002&pg=personal&fr_id=1000"}]}}
```

updateCaptainsMessage

Updates the team captain message for the team.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateCaptainsMessage &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*

`][&sign_redirects=`*value*`][&source=`*value*`][&sub_source=`*value*`][&success_redirect=`*value*`][&suppress_response_codes=`*value*`] &captains_message=`*value* `&fr_id=`*value*`[&auth=`*value*`][&sso_auth_token=`*value*`]`

Usage Notes

This method requires authentication. The logged-in user must be a Team Captain to execute this method successfully.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

captains_message

Required. The captain's message shown to all team members.

Type xsd:string.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateCaptainsMessageResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
</updateCaptainsMessageResponse>
```

JSON response

```
{"updateCaptainsMessageResponse":{"success":"true"}}
```

updateCompanyPageInfo

Updates a company page.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateCompanyPageInfo &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&company_id=`*value* `&fr_id=`*value* [`&headline1=`*value*][`&headline2=`*value*][`&headline3=`*value*][`&page_title=`*value*][`&rich_text=`*value*][`&status1=`*value*][`&status2=`*value*][`&status3=`*value*][`&status4=`*value*][`&status5=`*value*][`&text1=`*value*][`&text2=`*value*][`&text3=`*value*][`&auth=`*value*][`&sso_auth_token=`*value*]

Usage Notes

This method requires authentication, and must include the company_id. Only the logged-in constituent who is identified as the Company Coordinator in the Company record has permission to execute this method.

The Company Coordinator is defined by the Convio system administrator when the Company record is created.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

company_id

Required. The unique ID of the company.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

headline1

Optional. The first headline on the page.

Type xsd:string.

headline2

Optional. The second headline on the page.

Type xsd:string.

headline3

Optional. The third headline on the page.

Type xsd:string.

page_title

Optional. The title of the page

rich_text

Optional. The rich text area of the page.

Type xsd:string.

status1

Optional. Status Indicator 1

Type xsd:string.

status2

Optional. Status Indicator 2

Type xsd:string.

status3

Optional. Status Indicator 3

Type xsd:string.

status4

Optional. Status Indicator 4

Type xsd:string.

status5

Optional. Status Indicator 5

Type xsd:string.

text1

Optional. The first text area on the page.

Type xsd:string.

text2

Optional. The second text area on the page.

Type xsd:string.

text3

Optional. The third text area on the page.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateCompanyPageResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
</updateCompanyPageResponse>
```

JSON response

```
{"updateCompanyPageResponse":{"success":"true"}}
```

updateCompanyShortcut

Updates a team shortcut URL.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateCompanyShortcut &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&company_id=`*value* `&fr_id=`*value* [`&text=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method requires authentication, and must include the company_id. Only the logged-in constituent who is identified as the Company Coordinator in the Company record has permission to execute this method.

The Company Coordinator is defined by the Convio system administrator when the Company record is created.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

company_id

Required. The unique ID of the company.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

text

Optional. The shortcut text.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<updateCompanyShortcutResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <shortcutItem>
    <defaultUrl>http://example.com/teampageurl</defaultUrl>
    <hint>Sample hint text.</hint>
    <prefix>http://example.com/goto/</prefix>
    <text>sampleuser</text>
    <url>http://example.com/goto/sampleuser</url>
  </shortcutItem>
</updateCompanyShortcutResponse>
```

JSON response

```
{"updateCompanyShortcutResponse":{"shortcutItem":
{"text":"sampleuser","prefix":"http://example.com/goto/","hint":"Sample
 hint text.","defaultUrl":"http://example.com/teampageurl","url":"http://
example.com/goto/sampleuser"}}}
```

updateDraft

Saves a draft

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateDraft &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&fr_id=`*value* `&layout_id=`*value* `&message_id=`*value* `&message_name=`*value* `&subject=`*value* [`&message_body=`*value*][`&prepend_salutation=`*value*][`&recipients=`*value*][`&auth=`*value*][`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

layout_id

Required. The layout ID.

Type xsd:nonNegativeInteger.

message_id

Required. Message ID

Type xsd:nonNegativeInteger.

message_name

Required. The name of the message.

Type xsd:string.

subject

Required. The message subject.

Type xsd:string.

message_body

Optional. The message body.

Type xsd:string.

prepend_salutation

Optional. Setting to prepend personalized salutation to TAF emails

Type xsd:boolean.

recipients

Optional. The message recipients list.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateDraftResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <message>
    <messageId>2341</messageId>
    <subject>example subject</subject>
    <messageBody>example message body</messageBody>
    <messageName>example message name</messageName>
    <layoutId>1234</layoutId>
  </message>
</updateDraftResponse>
```

JSON response

```
{"updateDraftResponse":{"message":{"messageBody":"example message
 body","subject":"example
 subject","layoutId":"1234","messageId":"2341","messageName":"example message
 name"}}}
```

updatePersonalMediaLayout

Updates the personal media layout for a participants personal page

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updatePersonalMediaLayout &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&personal_media_layout=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

personal_media_layout

Optional. Media layout for a participant's personal page

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updatePersonalMediaLayoutResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <personalMediaLayout xsi:nil="true"/>
</updatePersonalMediaLayoutResponse>
```

JSON response

```
{"updatePersonalMediaLayoutResponse":{"personalMediaLayout":null}}
```

updatePersonalPageInfo

Updates a personal page.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updatePersonalPageInfo &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&blog_enabled=`*value*][`&headline1=`*value*][`&headline2=`*value*][`&headline3=`*value*][`&page_layout=`*value*][`&page_title=`*value*][`&rich_text=`*value*][`&status1=`*value*][`&status2=`*value*][`&text1=`*value*][`&text2=`*value*][`&text3=`*value*][`&auth=`*value*][`&sso_auth_token=`*value*]

Usage Notes

The page_title parameter sets the top heading for the personal page. The rich_text parameter sets the main page content. Only a subset of safe HTML tags are supported within rich_text: p, strong, em, span, ol, li, br, ul, and blockquote.

The blog_enabled parameter enables and disables blogging on the page.

The following values are meaningful in the status1 and status2 parameters:

- none_selected - no component
- participant_therm - a thermometer showing personal fundraising progress
- top_gifts_list - a scrolling list of top donors and their gifts
- donor_list - a scrolling list of donors with no amounts

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

blog_enabled

Optional. Whether or not your personal page blog is enabled.

Type xsd:boolean.

headline1

Optional. The first headline on the page.

Type xsd:string.

headline2

Optional. The second headline on the page.

Type xsd:string.

headline3

Optional. The third headline on the page.

Type xsd:string.

page_layout

Optional. The selected page layout

page_title

Optional. The title of the page

rich_text

Optional. The rich text area of the page.

Type xsd:string.

status1

Optional. Status Indicator 1

Type xsd:string.

status2

Optional. Status Indicator 2

Type xsd:string.

text1

Optional. The first text area on the page.

Type xsd:string.

text2

Optional. The second text area on the page.

Type xsd:string.

text3

Optional. The third text area on the page.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updatePersonalPageResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
</updatePersonalPageResponse>
```

JSON response

```
{"updatePersonalPageResponse":{"success":"true"}}
```

updatePersonalPagePrivacy

Update the privacy setting for the personal page.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updatePersonalPagePrivacy &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&is_private=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

is_private

Optional. This indicates whether the page is private or not.

Type xsd:boolean.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updatePersonalPagePrivacyResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <privatePage>true</privatePage>
</updatePersonalPagePrivacyResponse>
```

JSON response

```
{"updatePersonalPagePrivacyResponse":{"privatePage":"true"}}
```

updatePersonalVideoUrl

Saves the video URL for the user's personal page.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updatePersonalVideoUrl &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&fr_id=`*value* `&video_url=`*value* [`&auth=`*value*][`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

video_url

Required. The URL to the video.

Type xsd:anyURI.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updatePersonalVideoUrlResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
  <videoUrl xsi:nil="true"/>
</updatePersonalVideoUrlResponse>
```

JSON response

```
{"updatePersonalVideoUrlResponse":{"videoUrl":null}}
```

updateRegistration

Updates a registration record.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateRegistration` `&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&checkin_status=`*initial* | *started* | *committed* | *paid* | *complete* | *unknown* `&emergency_name=`*value* `&emergency_phone=`*value* `&flow_step=`*value* `&fr_id=`*value* `&goal=`*value* `&private_page=`*value* `&receive_gift_notification=`*value* [`&anonymous_registration=`*value*] [`&company_id=`*value*] [`&company_name=`*value*] [`&screenname=`*value*] [`&standard_registration=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

checkin_status

Required. Your check-in status.

Type xsd:string.

Options are:

- initial
- started
- committed
- paid
- complete
- unknown

emergency_name

Required. Emergency contact name.

Type xsd:string.

emergency_phone

Required. Emergency contact phone number.

Type xsd:string.

flow_step

Required. External application state.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

goal

Required. Your personal fundraising goal, in cents.

Type xsd:string.

private_page

Required. Whether or not your personal page should be hidden from all searches.

Type xsd:boolean.

receive_gift_notification

Required. Whether or not you wish to receive gift notifications.

Type xsd:boolean.

anonymous_registration

Optional. Whether or not your registration is to be designated anonymous.

Type xsd:boolean.

company_id

Optional. The id of the company the participant is to be assigned to.

Type xsd:nonNegativeInteger.

company_name

Optional. The name of a new company to be added and assigned to the team.

Type xsd:string.

screenname

Optional. If specified, your screen name will take the place of your name on public facing pages.

Type xsd:string.

standard_registration

Optional. Whether or not your registration is to be a standard registration.

Type xsd:boolean.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateRegistrationResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
</updateRegistrationResponse>
```

JSON response

```
{"updateRegistrationResponse":{"success":"true"}}
```

updateShortcut

Updates a shortcut URL.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateShortcut`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* `&text=`*value* [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

text

Required. The shortcut text.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateShortcutResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <shortcutItem>
    <defaultUrl>http://example.com/personalpageurl</defaultUrl>
    <hint>Sample hint text.</hint>
    <prefix>http://example.com/goto/</prefix>
    <text>sampleuser</text>
    <url>http://example.com/goto/sampleuser</url>
  </shortcutItem>
</updateShortcutResponse>
```

JSON response

```
{"updateShortcutResponse":{"shortcutItem":
{"text":"sampleuser","prefix":"http://example.com/goto/","hint":"Sample hint
 text.","defaultUrl":"http://example.com/personalpageurl","url":"http://
example.com/goto/sampleuser"}}}
```

updateSurveyResponses

Updates survey responses.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateSurveyResponses &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&question_=`*value*] [`&question_key_=`*value*] [`&require_response=`*value*] [`&survey_id=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Usage Notes

This method allows for updating a constituent's response to survey questions that they may or may not have previously taken. This is used primarily in the case of TeamRaiser surveys when the option to allow constituents to update their responses to registration questions is enabled within that TeamRaiser.

Note: This method does not enforce that option being enabled for the teamraiser specified by the fr_id.

Only questions that should normally be requested of the user will be updated. This means that if your TeamRaiser survey contains questions for All Participants, questions for Team Captains Only, and questions for only certain Participation Types, a participant who is not a Team Captain will only be allowed to provide updates to the questions for All Participants and their Participation Type.

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

question_

Optional. Include either question_ or question_key_ parameters for each question on the survey. Provide either existing values returned by the getSurveyResponses method or updated values. For questions with IDs 123 and 456, include parameters question_123 and question_456.

Type xsd:string.

question_key_

Optional. Include either question_ or question_key_ parameters for each question on the survey. Provide either existing values returned by the getSurveyResponses method or updated values. For questions with key 'T-Shirt_Size' and 'Permission_Slip_Returned', include parameters question_key_T-Shirt_Size and question_key_Permission_Slip_Returned.

Type xsd:string.

require_response

Optional. Verify that answers to questions requiring a response are not blank.

Type xsd:boolean.

Default is false.

survey_id

Optional. The survey ID (default is the event registration survey).

Type xsd:nonNegativeInteger.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateSurveyResponsesResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
</updateSurveyResponsesResponse>
```

JSON response

```
{"updateSurveyResponsesResponse":{"success":"true"}}
```

updateTeamInformation

Updates team related information. The company id and company name cannot be both passed since a team can only be assigned to a single company.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateTeamInformation &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&company_id=`*value*] [`&company_name=`*value*] [`&division_name=`*value*] [`&password=`*value*] [`&recruiting_goal=`*value*] [`&team_goal=`*value*] [`&team_name=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

company_id

Optional. The id of the company the team is to be assigned to.

Type xsd:nonNegativeInteger.

company_name

Optional. The name of a new company to be added and assigned to the team.

Type xsd:string.

division_name

Optional. The new team division name.

Type xsd:string.

password

Optional. The new team password.

Type xsd:string.

recruiting_goal

Optional. The new team recruiting goal.

Type xsd:string.

team_goal

Optional. The new team goal of the team, in cents.

Type xsd:nonNegativeInteger.

team_name

Optional. The new team name of the team.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateTeamInformationResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <companyName>example company name</companyName>
  <companyId>5</companyId>
  <teamName>example team name</teamName>
</updateTeamInformationResponse>
```

JSON response

```
{"updateTeamInformationResponse":{"teamName":"example team
 name","companyId":"5","companyName":"example company name"}}
```

updateTeamPageInfo

Updates a team page.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateTeamPageInfo &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&description=`*value*][`&rich_text=`*value*][`&auth=`*value*][`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

description

Optional. Description

Type xsd:string.

rich_text

Optional. The rich text area of the page.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateTeamPageResponse xsi:schemaLocation="http://convio.com/crm/v1.0 http://
service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://convio.com/crm/
v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <success>true</success>
</updateTeamPageResponse>
```

JSON response

```
{"updateTeamPageResponse":{"success":"true"}}
```

updateTeamShortcut

Updates a team shortcut URL.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateTeamShortcut`
`&api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*]
`&fr_id=`*value* [`&text=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

text

Optional. The shortcut text.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateTeamShortcutResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <shortcutItem>
    <defaultUrl>http://example.com/teampageurl</defaultUrl>
    <hint>Sample hint text.</hint>
    <prefix>http://example.com/goto/</prefix>
    <text>sampleuser</text>
    <url>http://example.com/goto/sampleuser</url>
  </shortcutItem>
</updateTeamShortcutResponse>
```

JSON response

```
{"updateTeamShortcutResponse":{"shortcutItem":
{"text":"sampleuser","prefix":"http://example.com/goto/","hint":"Sample
 hint text.","defaultUrl":"http://example.com/teampageurl","url":"http://
example.com/goto/sampleuser"}}}
```

updateTeamraiserAddressBookContact

Updates a contact within the context of teamraiser event. This allows you to update team address book contacts OR personal address book contacts.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateTeamraiserAddressBookContact &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&contact_id=`*value* `&email=`*value* `&first_name=`*value* `&fr_id=`*value* `&last_name=`*value* [`&city=`*value*] [`&country=`*value*] [`&county=`*value*] [`&phone=`*value*] [`&state=`*value*] [`&street1=`*value*] [`&street2=`*value*] [`&street3=`*value*] [`&zip=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

contact_id

Required. The system-generated unique contact ID for the contact.

Type xsd:nonNegativeInteger.

email

Required. The contact's email address.

Type xsd:string.

first_name

Required. The contact's first name.

Type xsd:string.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

last_name

Required. The contact's last name.

Type xsd:string.

city

Optional. The contact's city.

Type xsd:string.

country

Optional. The contact's country.

Type xsd:string.

county

Optional. The contact's county.

Type xsd:string.

phone

Optional. The contact's phone number.

Type xsd:string.

state

Optional. The contact's state or province.

Type xsd:string.

street1

Optional. The first line of the contact's street address.

Type xsd:string.

street2

Optional. The second line of the contact's street address, if needed.

Type xsd:string.

street3

Optional. The third line of the contact's street address, if needed.

Type xsd:string.

zip

Optional. The contact's postal or zip code.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateTeamraiserAddressBookContactResponse xsi:schemaLocation="http://
convio.com/crm/v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd"
 xmlns="http://convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <contact_id>1234</contact_id>
</updateTeamraiserAddressBookContactResponse>
```

JSON response

```
{"updateTeamraiserAddressBookContactResponse":{"contact_id":"1234"}}
```

updateTeamraiserData

Updates a teamraiser event

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
None.	SRTeamraiserAPI	POST	No.

Server API Syntax

`https://`*secure2.convio.net/organization*`/site/SRTeamraiserAPI?method=updateTeamraiserData` `&api_key=`*value* `&login_name=`*value* `&login_password=`*value* `&v=`*value* [`&center_id=`*value*] [`&response_format=`*xml* | *json*] [`&source=`*value*] [`&sub_source=`*value*] [`&suppress_response_codes=`*value*] `&update_teamraiser=`*value*

Usage Notes

This method accepts a list of one or more TeamRaiser Event data blocks up to a system-configurable maximum of 1000 Events, and asynchronously updates the values of any other fields specified.

For the list of available fields refer to *addTeamraiserData*.

Input data is passed url-encoded as the value of the update_teamraiser parameter in the following format:

```
<updateTeamraiserRequest xmlns="http://convio.com/crm/v1.0">
  <teamraiserUpdate>
    <frId>1014</frId>
    <eventUserTitle>Awesome-athon</eventUserTitle>
  </teamraiserUpdate>
  <teamraiserUpdate>
    <frId>1015</frId>
    <eventLocalOfficeAttn>Tiffany</eventLocalOfficeAttn>
  </teamraiserUpdate>
  <teamraiserUpdate>
    <frId>1016</frId>
    <eventLocalOfficeStreet1>504 Pecan St.</eventLocalOfficeStreet1>
  </teamraiserUpdate>
</updateTeamraiserRequest>}
```

The `<updateTeamraiserRequest>` document contains a collection of one or more `<teamraiserUpdate>` elements for each TeamRaiser event. Each block must contain a `<frId>` element specifying the record to update.

If the call is successful, it returns a unique job ID for the batch.

Server API Parameters

Common Server Parameters

See topic *Common Parameters*.

update_teamraiser

Required. XML document specifying the TeamRaiser events to update.

Type crm:updateTeamraiserList.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateTeamraiserResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <jobId>1359890305490</jobId>
</updateTeamraiserResponse>
```

JSON response

```
{"updateTeamraiserResponse":{"jobId":"1359890305490"}}
```

updateTentingStatus

Updates the tenting status for a participant.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=updateTentingStatus &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* `&tentmate_id=`*value* `&update_type=`*random* | *decline* | *accept* | *invite* | *reset* | *reject* [`&message=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

tentmate_id

Required. The constituent id of the selected tentmate.

Type xsd:nonNegativeInteger.

update_type

Required. This specifies the type of update to be made.

Type xsd:string.

Options are:

- random
- decline
- accept
- invite
- reset
- reject

message

Optional. The message content that will be sent. This only applies to invite, accept, and reject type updates.

Type xsd:string.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<updateTentingStatusResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <tentmateStatus>1</tentmateStatus>
</updateTentingStatusResponse>
```

JSON response

```
{"updateTentingStatusResponse":{"tentmateStatus":"1"}}
```

uploadCompanyPhoto

Uploads a company page photo. If the graphic_upload_file is blank, the photo caption will be the only item updated. The photo must be submitted in encoding type 'multipart/form-data'.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=uploadCompanyPhoto &api_key=`*value* `&v=`*value* [`&auth=`*value*][`&center_id=`*value*][`&error_redirect=`*value*][`&redirect=`*value*][`&response_format=`*xml* | *json*][`&sign_redirects=`*value*][`&source=`*value*][`&sub_source=`*value*][`&success_redirect=`*value*][`&suppress_response_codes=`*value*] `&company_id=`*value* `&fr_id=`*value* [`&photo1_text=`*value*][`&photo_1_file=`*value*][`&photo_1_upload=`*value*][`&auth=`*value*][`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

company_id

Required. The unique ID of the company.

Type xsd:nonNegativeInteger.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

photo1_text

Optional. The photo caption for the team photo

photo_1_file

Optional. The name of the team photo to be uploaded.

photo_1_upload

Optional. Indicates whether team photo should be updated.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<uploadCompanyPhotoResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <photoItem>
    <id>123</id>
    <thumbnailUrl>http://www.example.com/thumbnailexample.jpg</thumbnailUrl>
  </photoItem>
</uploadCompanyPhotoResponse>
```

JSON response

```
{"uploadCompanyPhotoResponse":{"photoItem":{"id":"123","thumbnailUrl":"http://
www.example.com/thumbnailexample.jpg"}}}
```

uploadPersonalPhoto

Uploads a personal page photo. If the graphic_upload_file is blank, the photo caption will be the only item updated. The photo must be submitted in encoding type 'multipart/form-data'.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=uploadPersonalPhoto &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&graphic_caption=`*value*] [`&graphic_caption2=`*value*]

[`&graphic_upload2_file=`*value*] [`&graphic_upload2_upload=`*value*] [`&graphic_upload_file=`*value*] [`&graphic_upload_upload=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

graphic_caption

Optional. The new caption for photo 1 to be updated.

graphic_caption2

Optional. The new caption for photo 2 to be updated.

graphic_upload2_file

Optional. The file representing the personal photo 2 to be uploaded.

graphic_upload2_upload

Optional. Indicates whether photo 2 should be updated.

graphic_upload_file

Optional. The file representing the new personal photo 1 to be uploaded.

graphic_upload_upload

Optional. Indicates whether photo 1 should be updated.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<uploadPersonalPhotoResponse xsi:schemaLocation="http://convio.com/crm/
v1.0 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <photoItem>
    <id>123</id>
    <thumbnailUrl>http://www.example.com/thumbnailexample.jpg</thumbnailUrl>
  </photoItem>
</uploadPersonalPhotoResponse>
```

JSON response

```
{"uploadPersonalPhotoResponse":{"photoItem":
{"id":"123","thumbnailUrl":"http://www.example.com/thumbnailexample.jpg"}}}
```

uploadTeamPhoto

Uploads a team page photo. If the graphic_upload_file is blank, the photo caption will be the only item updated. The photo must be submitted in encoding type 'multipart/form-data'.

Client Servlet	Server Servlet	HTTP Methods Supported	Requires Authentication
CRTeamraiserAPI	None.	POST	Yes.

Client API Syntax

`https://`*secure2.convio.net/organization*`/site/CRTeamraiserAPI?method=uploadTeamPhoto &api_key=`*value* `&v=`*value* [`&auth=`*value*] [`&center_id=`*value*] [`&error_redirect=`*value*] [`&redirect=`*value*] [`&response_format=`*xml* | *json*] [`&sign_redirects=`*value*] [`&source=`*value*] [`&sub_source=`*value*] [`&success_redirect=`*value*] [`&suppress_response_codes=`*value*] `&fr_id=`*value* [`&graphic_caption=`*value*] [`&graphic_upload_file=`*value*] [`&graphic_upload_upload=`*value*] [`&auth=`*value*] [`&sso_auth_token=`*value*]

Client API Parameters

Common Client Parameters

See topic *Common Parameters*.

fr_id

Required. The TeamRaiser event ID.

Type xsd:nonNegativeInteger.

graphic_caption

Optional. The photo caption for the team photo

graphic_upload_file

Optional. The name of the team photo to be uploaded.

graphic_upload_upload

Optional. Indicates whether team photo should be updated.

auth

Optional. You must provide either this parameter or the sso_auth_token parameter. If using this parameter, you must specify an auth token generated by an embedded [[S86]] session tag in a Convio web page.

sso_auth_token

Optional. You must provide either this parameter or the auth parameter. If using this parameter, you must specify a valid session authentication token returned by *login*, *authenticateUser*, or *getSingleSignOnToken*.

Response

See topic *HTTP Status Codes*.

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<uploadTeamPhotoResponse xsi:schemaLocation="http://convio.com/crm/v1.0
 http://service.convio.net/xmlschema/crm.public.v1.xsd" xmlns="http://
convio.com/crm/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <photoItem>
    <id>123</id>
    <thumbnailUrl>http://www.example.com/thumbnailexample.jpg</thumbnailUrl>
  </photoItem>
</uploadTeamPhotoResponse>
```

JSON response

```
{"uploadTeamPhotoResponse":{"photoItem":{"id":"123","thumbnailUrl":"http://
www.example.com/thumbnailexample.jpg"}}}
```