Table of Contents

Quick Base HTTP API Reference

The HTTP API is a collection of calls for accessing Quick Base applications over the web. It is a RESTful API over HTTP with XML payloads and an interface only; no components, modules, or libraries are installed at the application end.

Using the Quick Base API, you can:

- Provide secure access to Quick Base and Quick Base applications.
- Manage access through the Quick Base roles feature.
- Automate the application lifecycle: creation, deletion, and copying (cloning) of applications.
- Create, modify, and delete the application's tables and fields
- Add, update, delete, and querying table records

Find answers

- **Search.** Use the **Search** field in the upper right corner of the page to search within the API Guide. Use the in-product **Help** button on the Quick Base Global bar to search all help resources.
- **Index.** Help also contains an index to help you find specific topics. Click the **Index** tab in the left-hand pane, and type a word in the **Search Index** box.
- **Table of Contents.** Click the **Contents** tab in the left hand pane to browse Quick Base topics.
- **Quick Base Online Help.** Click the **Online Help** button in the toolbar above the right-hand pane to jump to topics that explain all of the features available with Quick Base.

In this guide

This guide contains these sections:

- Overview – Explains the basics about how the API works and how to set up and authenticate users using the API.
- Quick Base API Call Reference – Describes each API call in detail (these are listed alphabetically.)
- Appendix A: Error Codes – Lists and describes the messages Quick Base returns when it encounters an error.
- Appendix B: Managing Files – Provides an overview of uploading and downloading file attachments
- Appendix C: Building Queries – Provides examples of how to use the different components that make up a query.
- Appendix D: APIs and International Use of Quick Base – Provides information on API calls that are affected by changes to Quick Base's currency symbols, their placement relative to a number, number formats, and date formats.

Did this help you? Give us a rating:

Before you begin: User setup

Before you can allow a user to access Quick Base through the API, you need to provision the user in Quick Base. Provisioned users have the following:

- A Quick Base username and password. Typically, users set up their own usernames and passwords when they've been invited to a Quick Base application.
- A Quick Base role. Quick Base roles define a user's level of access. You'll need to set up roles for your application using the Quick Base UI. Once you've established roles, you can assign them to users using either the Quick Base UI or the API.

Once you've provisioned a user in Quick Base, they can authenticate and access Quick Base through the API.

Note: If you use the Quick Base API to submit data to your app via a web form, you must ensure that the role used to submit the data has at least Basic Access application permissions. If the app level permission is set to *None*, you will receive an error message when data is submitted. Note that this includes scenarios where a "user agent" submits data on behalf of anonymous users, or where the data is submitted through the "Everyone on the Internet" persona.

Did this help you? Give us a rating:

Authentication and secure access

Once you've authenticated as a Quick Base user, you can invoke API calls to access Quick Base data. Depending on how your app and account are set up, you may be required to supply the following with each API call:

- An authentication ticket or user token – to authenticate yourself to Quick Base. The user token is an alternate means of authentication for APIs and automation that is generally more secure and convenient. Upon receiving the API call, Quick Base examines the authentication and application token you provided.
- An application token for the app that the API call will access – some apps require an application token to be provided in addition to a means of authentication. An application token is an extra string of characters used to verify that access to a Quick Base application is authorized.

- **Authenticating with a ticket:** If the ticket is valid, the API call is allowed to proceed. If the app also requires an application token, that is also checked before the API call can proceed.
- **Authenticating with a user token:** If you authenticate yourself to Quick Base with a user token, no application token is needed, even if one is assigned to the app you're accessing. The user token can be assigned to one or more apps, and provides built-in security that ticket authentication does not. However, user tokens are not allowed for all API calls.

About the authentication ticket

When you authenticate a user using API_Authenticate, Quick Base returns an authentication ticket in these two formats:

- a browser cookie--If you are using a web browser to access Quick Base, Quick Base uses the ticket cookie returned by the API_Authenticate call to authenticate users. If you are using a browser and have enabled cookies, Quick Base continues to use the cookie after the user has authenticated; you need not supply the cookie with each API call after you've authenticated the user with API_Authenticate.
- an XML representation of the ticket for inclusion in subsequent API calls--If you are NOT accessing Quick Base using a web browser or you've not enabled cookies, you must supply the XML representation of the ticket in each API call. (Do NOT hard code the username and password parameter and their values in your request, even if you are using https.)

By default, the ticket is valid for 12 hours. You can change this default in the API_Authenticate call; you can set the duration of the ticket's validity from one hour to several days.

Note: API_Authenticate calls have a maximum time limit of 4,380 hours, or approximately 6 months. This applies to new tickets initiated on or after January 21, 2018.

Many API calls that use a ticket can instead take the usertoken parameter. You can still use the ticket method if you prefer. The usertoken parameter can be used with any API that doesn't post to a db/main URL, and also with API_GrantedDBs.

Supplying the authentication ticket and application or user tokens in API calls

The following table describes when you need to supply the authentication ticket, application token, or user token. It assumes you've already authenticated the user using API_Authenticate (or are using a user token).

ARE YOU ACCESSING QUICK BASE VIA A WEB BROWSER (CAN YOU RELY ON COOKIES)?	ARE APPLICATION OR USER TOKENS REQUIRED?	YOU MUST SUPPLY THE FOLLOWING IN EACH API CALL:
Yes	Yes	The application token only. You need not supply the authentication ticket explicitly; the authentication ticket is supplied automatically by the cookie.
No	Yes	The authentication ticket, application token, or user token
Yes	No	Neither the authentication ticket, application token, or user token. The app or user token is not required, and, if you are using a web browser, the authentication ticket will be supplied automatically.
No	No	The authentication ticket or user token

Did this help you? Give us a rating:

About user tokens

You can create your own user tokens in Quick Base and use them to run APIs and automation with your permissions.

Many API calls that use a ticket can instead take a parameter called *usertoken*. You can still use the ticket method, if you prefer. You can use any user token parameter with any API that doesn't post to a db/main URL, and also with API_GrantedDBs.

After you have created one or more user tokens, you can activate, deactivate, or delete them. For more details about managing user tokens, click here.

Did this help you? Give us a rating:

Understanding the API call reference

The Quick Base API can be invoked in either of two ways.

- As an XML request string POSTed to Quick Base
- As a URL sent in a GET request (e.g. sent from a browser, or embedded in a web page or Word document).

Note that there are many samples throughout this guide for both XML and URL requests. The parts of these sample calls that reference specific information, such as authentication ticket, application token, user token, app or table dbid, and customer domain are represented by italicized sample text, as shown below:

- An authentication ticket - *auth_ticket*
- An application token - *app_token*
- A user token - *user_token*
- An app or table dbid - *target_dbid*
- A Quick Base domain addressed by the call - *target_domain*

For most API calls, Quick Base returns a response in XML format. For certain API calls that are embedded in a web page or Word document, Quick Base returns HTML responses.

Understanding the XML Samples

Each sample can be divided into two sections: the POST request method and the actual API call.

POST request method

An example of the POST request method appears below.

```
POST https://target_domain/db/target_dbid
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION:API_AddField
```

The first three lines shown above are standard HTTP headers for the POST request method. The first line specifies that the secure HTTPS protocol should be used and contains the URL for the POST. The URL includes the *target_domain*, or the domain against which you are invoking the call (for example, *quickbase.com*). The URL also includes the *target_ dbid*, which identifies the app or table the call should address.

The last line is a custom Quick Base header containing the API call.

API call

The second part of each sample illustrates the syntax for the API call. The elements between the `<qdbapi>` tags are the parameters; these are documented for each API call.

```
<qdbapi>
   <label>label</label>
   <type>type</type>
   <mode>mode</mode> <!-- optional unless lookup or formula-->
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <udata>whatever</udata> <!--optional pass-thru data -->
</qdbapi>
```

Order of parameters in API calls (XML or URL)

Quick Base accepts API call parameters in any order, whether the call is made via XML or a URL.

Using key fields

By default, each table has the built-in Record ID# (rid) field, which provides the unique key for that record within the table. You can use another field as the key field if you'd like, as long as the new field type supports the **Unique** property.

If you use a field other than the Record ID# as the key field, you may be able to replace the **&rid** parameter in many of the API calls with the **&key** parameter. The following API calls support this substitution:

- API_ChangeRecordOwner
- API_DeleteRecord
- API_EditRecord
- API_GetRecordAsHTML
- API_GetRecordInfo

Did this help you? Give us a rating:

API call reference by function

The following table groups the API calls under different functional categories.

FUNCTIONAL CATEGORIES	RELATED API CALLS AND FEATURES
Application and table metadata	• API_FindDBByName • API_GetAncestorInfo • API_GetAppDTMInfo • API_GetDBInfo • API_GetSchema • API_GrantedDBs
Creating, copying, and deleting applications	• API_CloneDatabase • API_CreateDatabase • API_DeleteDatabase • API_RenameApp

FUNCTIONAL CATEGORIES	RELATED API CALLS AND FEATURES
Uploading and downloading files	• API_AddRecord • API_EditRecord • API_UploadFile
Secure access to Quick Base	• API_Authenticate • API_SignOut
Table and field management	• API_AddField • API_CreateTable • API_DeleteField • API_FieldAddChoices • API_FieldRemoveChoices • API_SetFieldProperties • API_SetKeyField
Record (data) management	• API_AddRecord • API_ChangeRecordOwner • API_CopyMasterDetail • API_DeleteRecord • API_DoQuery • API_DoQueryCount • API_EditRecord • API_GenAddRecordForm • API_GenResultsTable • API_GetNumRecords • API_GetRecordAsHTML • API_GetRecordInfo • API_ImportFromCSV • API_PurgeRecords • API_RunImport

FUNCTIONAL CATEGORIES	RELATED API CALLS AND FEATURES
Managing user access	- API_AddUserToRole - API_ChangeUserRole - API_ChangeRecordOwner - API_ChangeManager - API_GetRoleInfo - API_GetUserInfo - API_GetUserRole - API_ProvisionUser - API_RemoveUserFromRole - API_SendInvitation - API_UserRoles
Managing groups	- API_AddGroupToRole - API_AddSubGroup - API_AddUserToGroup - API_ChangeGroupInfo - API_CopyGroup - API_CreateGroup - API_DeleteGroup - API_GetGroupRole - API_GetUsersInGroup - API_GrantedDBsForGroup - API_GrantedGroups - API_RemoveGroupFromRole - API_RemoveSubgroup - API_RemoveUserFromGroup
Miscellaneous functions	- API_AddReplaceDBPage - API_GetDBPage - API_GetDBvar - API_SetDBvar

FUNCTIONAL CATEGORIES	RELATED API CALLS AND FEATURES
Webhooks	• API_Webhooks_Activate • API_Webhooks_Copy • API_Webhooks_Create • API_Webhooks_Deactivate • API_Webhooks_Delete • API_Webhooks_Edit

Did this help you? Give us a rating:

Common optional parameters

The documentation for each API call lists the parameters you can use with each call. Some parameters can be used with many calls. These are listed in the table below.

PARAMETER	VALUE	DESCRIPTION
encoding	A string that specifies the encoding parameter of the XML output	The Quick Base API outputs XML using the Unicode character set. If you are using Shift-JIS as your character encoding for a particular Quick Base table, you should specify the encoding parameter on every call you make to ensure that the character encoding of the XML response is properly set. For example: `<?xml version="1.0" encoding="shift_jis" ?>`

PARAMETER	VALUE	DESCRIPTION
rdr	A redirect URL that should be used when there is no error	Allows you to specify a redirect URL. Note that the URL you specify here is not used when Quick Base encounters an error. In error situations, Quick Base always redirects to a standard error page, regardless of what you specify here. If you don't specify a value here, Quick Base does not perform the redirect. Redirect URLs often include query strings with special characters that need to be properly encoded. Use **%3D** for the equals sign (=), and **%3F** for the question mark (?), and **%26** for the ampersand (&).

xsl	A URL that points to an XSL document that is referenced in the XML output from the call	Usually is omitted, in which case the XML return page does not include any reference to an XSL document.
		XSL URLs often have query strings in them that cause errors unless they are properly encoded. Ampersands have special significance in XML so they can't be used in URLs for XSL stylesheets. Quick Base understands carets (^) as a replacement for ampersands when delimiting name/value pairs in query strings. Encode the caret as **%5E** in API calls. For example:
		`?a=dbpage&pagename=mypage.xsl`
		becomes
		`%3Fact%3Ddbpage%5Epagename%3Dmypage.xsl`

PARAMETER	VALUE	DESCRIPTION

msInUTC	1—True. Date/time values are specified in milliseconds as UTC
	0 — False. Date/time values are specified in milliseconds using the local application time
		Allows you to specify that Quick Base should interpret all date/time stamps passed in as milliseconds using Coordinated Universal Time (UTC) rather than using the local application time.
		To understand how to use UTC, consider this example. In this example:
		• your billing account is set to use PST
		• you have a table where the field with ID 7 is a Date/Time field
		If you issue this API call without using the **msInUTC** parameter:
		`?a=API_AddRecord&_fid_7=1267642263823`
		…Quick Base interprets the value for field ID 7 as milliseconds in PST. If you then display the record you added, Quick Base displays this value: 03-03-2010 06:51 PM . (The display is always in application local time).
		If instead you include the **msInUTC** parameter and issue this API call:
		`?a=API_AddRecord&_fid_7=1267642263823&msInUTC=1`
		...Quick Base interprets the value for field ID 7 as milliseconds in UTC. If you then display the record you added, Quick Base displays this value:03-03-2010 10:51 AM

realmhost	A domain used to access Quick Base	If you use a domain other than *www.quickbase.com* to access Quick Base, you can use this parameter to specify it in your API calls. For example: `?a=API_Authenticate&username=PTBarnum&password=TopSecret &realmhost=`*target_domain*`.quickbase.com` where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*.

PARAMETER	VALUE	DESCRIPTION
X_QUICKBASE_RETURN_HTTP_ERROR	A header that causes Quick Base to return a 400 (Bad Request) status when there is an API error.	By default, APIs return an HTTP status of 200 if Quick Base is able to return any response, even if there's an API error. However, if the API call passes the header `X_QUICKBASE_RETURN_HTTP_ERROR=true`, Quick Base will return a 400 (Bad Request) status when there is an API error. The detailed error code and message will be in the response body.

Did this help you? Give us a rating:

API_AddField

Overview

Use API_AddField to add a new field to a table. You invoke this call on a table-level dbid.

When you add a field using API_AddField, you specify the field type, but no other field properties. After you've added the field, you can use API_SetFieldProperties to set the properties of the new field and any default values. (You can't set field type using API_SetFieldProperties; if you want to change the field type after adding the field, you must use the Quick Base UI.)

The amount of data space consumed by a field depends on the field type. Read more in the online help.

Request parameters

PARAMETER	VALUE	REQUIRED?
add_to_forms	Specifies whether the field you are adding should appear at the end of any form with form properties set to "Auto-Add new fields." Set this parameter to 1 if you want the field to appear on all forms with "Auto-Add new fields" enabled. If the new field has a type of fkey or dblink, the form is set to display it as a report instead of as a link.	no
apptoken	A valid application token.	yes, if the application requires application tokens
label	Allows you to enter the name of the new field.	yes

PARAMETER	VALUE	REQUIRED?
mode	Specifies whether the field is a formula field or a lookup field. Valid values are: • virtual -- specifies that the field should be a formula field (of whatever type you specify). You can specify virtual mode for any field type. • lookup -- Specifies that the field is a Lookup field. You can specify lookup mode for text (Text) or float (Numeric) type fields. The value for this parameter must be lowercase.	Required for lookup or formula fields
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	user token	yes, one of: • ticket • username/password • user token

<table>
<tr>
<td>type</td>
<td>The Quick Base field type. The eligible type names differ slightly from their counterparts in the Quick Base UI:
<table>
<tr><th>QUICK BASE UI: FIELD TYPE</th><th>QUICK BASE API: FIELD TYPE</th></tr>
<tr><td>Checkbox</td><td>checkbox</td></tr>
<tr><td>Date</td><td>date</td></tr>
<tr><td>Duration</td><td>duration</td></tr>
<tr><td>Email Address</td><td>email</td></tr>
<tr><td>File Attachment</td><td>file</td></tr>
<tr><td>Formula</td><td>(see the “mode” param)</td></tr>
<tr><td>Lookup</td><td>(see the “mode” param)</td></tr>
<tr><td>List - User</td><td>multiuserid</td></tr>
<tr><td>Multi-select Text</td><td>multitext</td></tr>
<tr><td>Numeric</td><td>float</td></tr>
<tr><td>Numeric - Currency</td><td>currency</td></tr>
<tr><td>Numeric - Percent</td><td>percent</td></tr>
<tr><td>Numeric - Rating</td><td>rating</td></tr>
<tr><td>Phone Number</td><td>phone</td></tr>
<tr><td>Report Link</td><td>dblink</td></tr>
</table>
</td>
<td>yes</td>
</tr>
</table>

<table>
<tr><th>PARAMETER</th><th>VALUE</th><th>REQUIRED?</th></tr>
<tr><td></td><td><table><tr><td>Text</td><td>text</td></tr><tr><td>Time Of Day</td><td>timeofday</td></tr><tr><td>URL</td><td>url</td></tr><tr><td>User</td><td>userid</td></tr></table></td><td></td></tr>
<tr><td>udata</td><td>A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.</td><td>no</td></tr>
</table>

Response values

The response to this call contains the following:

ELEMENT NAME	VALUE
action	The originating request, for example, API_AddField.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
fid	The field ID. You can use this value in other API calls that manipulate this field or field values.

ELEMENT NAME	VALUE
label	The name of the field as it appears in the Quick Base application UI.
udata	Optional. Contains any udata value supplied in the request.

Sample XML request

```
POST https://target_domain/db/target_dbid?
 Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_AddField

<qdbapi>
  <label>Phone Number</label>
   <type>phone</type>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_AddField&label=Phone+Number
&type=phone&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_AddField</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <fid>8</fid>
   <label>Phone Number</label>
</qdbapi>
```

Did this help you? Give us a rating:

API_AddGroupToRole

Overview

Use API_AddGroupToRole to add a group to a role in an app.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the group to manage.	yes
roleid	The ID of the access role being assigned to the group.	yes
apptoken	A valid application token.	yes, if the application requires application tokens.
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_AddGroupToRole.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. A response containing "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid? HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_AddGroupToRole

<qdbapi>
   <udata>misc data</udata>
    <apptoken>app_token</apptoken>
   <ticket>auth_ticket</ticket>
   <gid>345889.ksld</gid>
   <roleid>12</roleid>
</qdbapi>
```

Sample HTTP Get Request

```
https://*target_domain*/db/*target_dbid*?a=API_AddGroupToRole&gid=345889.ksld&
roleid=12&ticket=*auth_ticket*&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_AddGroupToRole</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_AddRecord

Overview

Use API_AddRecord to add a record to a table. You invoke this call on a table dbid. When you use this call, you add fields and their values for the record. You must add values for all required fields or API_AddRecord returns an error.

The amount of data space consumed by a field depends on the field type. Read more in the online help.

Using field IDs and field names

When specifying fields, you can use either field IDs (fids) or field names. Field names are nearly identical to field labels; if you know the field label, you can determine the field name by converting all characters to lowercase and replacing all non-alphanumeric characters (including spaces) with an underscore.

For example, for a field with this label...

Event Name

...you should use this field name:

event_name

You can use a mixture of fids and field names in the same API call.

- To obtain a field name, first determine the field's label (using API_GetSchema, or by viewing the field in the Quick Base application UI.) You can derive the field name by replacing uppercase characters with lowercase characters and using underscores instead of all non-alphanumeric characters.
- To obtain a fid, use API_GetSchema. If you are an application manager, you can also obtain the fid using the Quick Base UI by customizing field properties.

What happens if I write to built-in fields?

Every record has the same built-in Quick Base fields. For example, every record contains a Record ID. If you write to one of these fields by mistake, you'll get an error and the record won't be added. (You can bypass this error using the **ignoreError** parameter in your API call.)

What happens if I write to non-writable fields?

Fields such as formula fields, the iCalendar field, and the vCard field are non-writable; that is, you cannot add data to these fields. If your API call writes to a non-writable field, the call is ignored.

vCard and iCalendar are widgets linked to fields in a table. If you want to add data to these, you must write to those table fields; the vCard and iCalendar fields will be updated with that data.

Can I add a record without supplying data for all fields?

You must supply field data for all required fields. You are not required to supply values for fields that are not required, but if you do not, those fields will default to the values specified in field properties.

What data validation is enforced on field data I write?

Quick Base validates data in some, but not all, cases. For example, in a field of type Email Address, you can supply a non-valid email address and Quick Base will add it. However, in other cases, Quick Base will not write invalid data to the record. For example, if you supply an invalid value for Numeric or Phone Number fields, Quick Base won't write the invalid data. But, unless these fields are required, Quick Base will add the record.

Request parameters

The following parameters are available:

PARAMETER	VALUE	REQUIRED?

field	Specify values for the fields that make up the record you want to add using either the **fid** attribute or **name** attribute of the `<field>` element.
	For example:
	`<field fid="18">Hi!</field>`
	`<field name="Message">Hi!</field>`
	You must specify all required fields.
	The values you enter here will vary, depending on field type. See How to specify values for different field types for more information.
		yes

disprec	Set this parameter to 1 to specify that the new record should be displayed within the Quick Base application. An application login required before the record can be displayed. If you use this parameter, Quick Base, returns the normal Quick Base HTML page that displays the record. Omit this property if you don't want the new record to display within the Quick Base application.	no

PARAMETER	VALUE	REQUIRED?
fform	Set this parameter to 1 if you are invoking API_AddRecord from within an HTML form that has checkboxes and want those checkboxes to set Quick Base Checkbox fields. Omit this property if you don't need Quick Base to set Checkbox fields based on your HTML page.	no
ignoreError	Set this parameter to 1 to specify that no error should be returned when a built-in field (for example, **Record ID#**) is written-to in an API_AddRecord call. If you do not set this parameter, Quick Base returns an error when API_AddRecord writes to a built-in field.	no
ticket	Specifies valid authentication ticket. You obtain a valid ticket from API_Authenticate. A ticket is also returned in the various API responses. See API Overview for more information on the authentication ticket.	No.
usertoken	A user token – this can be used for API access and is the preferred alternative to a ticket or application token.	yes, one of: • ticket • username/password • user token
apptoken	Specifies a valid application token. See API Overview for more information on application tokens.	yes, if the application requires application tokens and you are not using a user token
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

PARAMETER	VALUE	REQUIRED?
msInUTC	Allows you to specify that Quick Base should interpret all date/time stamps passed in as milliseconds using Coordinated Universal Time (UTC) rather than using the local application time. Set this parameter to 1 if you want to use Coordinated Universal Time. See usage example.	no
msAsDurationDefault	Set this parameter to 1 to specify milliseconds, instead of days, as the default units for a duration field value sent without any units. If you set this parameter to 0 or omit it, then any duration field values sent without a unit will default to being interpreted as days.	no
clist	A period-delimited list of field IDs to be returned. Quick Base will return values for these field IDs in the order in which you enter them here. To return all fields in a table, set this parameter to the value a. Allows you to return the state of dynamic fields like formulas, lookups or summaries without combining with API_DoQuery.	no

How to specify values for different field types

The following table describes how you can specify values for different field types. Note that, once you enter values using API_AddRecord, you can set properties for each field using API_SetFieldProperties.

FIELD TYPE	ACCEPTABLE VALUES
Text	Any characters, special characters, numbers, or symbols. Note that non-alphanumeric characters (anything other than A-Z, a-z, and 0-9) may need to be encoded to appear as intended in your data.

FIELD TYPE	ACCEPTABLE VALUES
Text - Multi-line	Any characters, special characters, numbers, or symbols. Note that non-alphanumeric characters (anything other than A-Z, a-z, and 0-9) may need to be encoded to appear as intended in your data.
Text - Multiple Choice	A valid choice that has been set up for the multiple choice field. Note that Quick Base does not validate case here; if you enter "ford" and, in your application, the choice is "Ford," the choice will be accepted. If you enter an invalid choice, Quick Base generates an error.
Multi-select Text	Up to 20 valid choices from the list set up for the field, separated by semi-colons. The set of choices provided must not contain duplicates. **Note:** Choices added to a Multi-select Text field are limited to 60 characters, and the total number of choices in the field may not exceed 100.
Numeric	Positive or negative numbers. Quick Base ignores any non-numeric characters you enter here, but will not generate an error. If you've specified decimal places using API_SetFieldProperties, the value will be truncated or lengthened accordingly.
Numeric - Currency	Positive and negative numbers, with or without decimals. The decimal character should match the decimal character set in the field's properties.
Numeric - Percent	A number that represents the percentage. Note that, if you want to indicate 80%, you should enter 80 in this field.
Numeric - Rating	A numeric rating, from 1 - 5. Quick Base displays ratings as stars; if you enter 3 in a Numeric-Rating field, Quick Base displays 3 out of 5 stars selected.

FIELD TYPE	ACCEPTABLE VALUES
Date	A date in the format specified in the app's properties. Alternatively, a date in milliseconds since January 1, 1970 00:00:00 UTC. Note that the Quick Base HTTP API returns dates in this format, which is the same internal representation used by JavaScript.
Date/Time	A date and time. Dates should be in the format specified in the app's properties. This field is an extended date field that can also contain the time, in the format HH:MM AM/PM. If you don't specify AM or PM, Quick Base defaults to AM. If you don't specify a time, Quick Base defaults to 12:00 AM.
Time of Day	A time in this format: HH:MM AM/PM. If you do not specify AM or PM, Quick Base defaults to AM.
Duration	A number that indicates a period of time. Note that you must use API_SetFieldProperties to set the unit of measure. If you enter a non-numeric value here, Quick Base ignores the value (no error is generated.)
Checkbox	A string that indicates whether the checkbox is checked or not. To specify that a checkbox is checked, enter any of these values: • 1 • yes • true • on To specify that a checkbox is not checked, enter any string other than those listed above, or leave the parameter blank. If a Checkbox field is required, and does not default to "checked," you must enter some value to be able to save the record.

FIELD TYPE	ACCEPTABLE VALUES
Phone Number	A phone number, with or without an extension. Enter a 10-digit string of numbers. You are not required to enter special characters (parentheses or dashes). Example: For this phone number: (123) 456-7890 ...enter 1234567890 If you want to include an extension, you can enter x after the last digit of the phone number, followed by the numeric characters that make up the extension. There is no minimum or maximum character limit on extensions. Example: For this phone number:(123) 456-7890 x9876 ...enter 1234567890x9876 Quick Base ignores any non-numeric character you enter here (except for the x used for extensions).
Email Address	An email address (joeuser@example.com). Note that if you enter an invalid email address, Quick Base does not generate an error.
User	A Quick Base user's email address or Quick Base user name.
List-User	Quick Base users' email addresses, Quick Base user names, or hashed user IDs, separated by semi-colons. Example: joe@example.com;sue@example.com

FIELD TYPE	ACCEPTABLE VALUES
File Attachment	A base64-encoded file. See Managing Files for more information about uploading files. Note that you must not use MIME encoding and must not include MIME headers. Note that many base64 encoders or base64 encoding methods are for MIME type encoding and will not work with Quick Base. You must not insert any newline characters when you encode the file. If your file attachments appear to upload but don't display in Quick Base, double check for these characters. In the opening <field> tag, insert the **filename** attribute in addition to the **fid** or **name** attribute, as shown in this example: `<field fid="22" filename="profilePhoto.jpg">` The **filename** attribute value should be set to the name of the file with no path specified. Insert the base64-encoded text from the encoded file between the opening and closing <field> tags.
URL	A Web address. If you don't enter "http://", Quick Base adds it for you. Note that if you enter an invalid Web address, Quick Base does not generate an error.
Report Link	Report links are derived from other fields. You can update which report is linked to by updating the field that the report link refers to. You can't write to this type of field directly. If your API writes to a Report Link field, Quick Base ignores the call.
iCalendar	iCalendar fields are derived from other fields. You can update this type of field only by updating the fields to which it refers. You can't write to this type of field directly. If your API writes to an iCalendar field, Quick Base returns an error.
vCard	vCard fields are derived from other fields. You can update this type of field only by updating the fields to which it refers. You can't write to this type of field directly If your API writes to a vCard field, Quick Base returns an error.

FIELD TYPE	ACCEPTABLE VALUES
Predecessor	The Record ID of the predecessor record. Note that if you enter an invalid Record ID here, Quick Base returns an error.
Formula fields	Formula fields are derived from other fields. You cannot write to this type of field directly. If your API writes to a formula field, Quick Base returns an error.

Response values

The response to this call contains the following:

ELEMENT NAME	VALUE
action	The originating request, for example, API_AddRecord.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
rid	Record ID of the record that was added
update_id	Used to detect update conflicts when invoking API_EditRecord. You could save this update ID when you add a new record, but it would be better to instead get the most recent update_id value later when you query for the record to get it and update it.

Sample XML Request

```
POST https://target_domain/db/target_dbid?
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_AddRecord
```

Example Using Field Names

```
<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <field name="event_name">party at Lindisfarne</field>
   <field name="description">dress in style of the epoch</field>
   <field name="location">lindisfarne island</field>
   <field name="start_time">06-08-0793</field>
   <field name="end_time">10-14-1066</field>
</qdbapi>
```

Example Using Field IDs

```
<qdbapi>
<udata>mydata</udata>
<ticket>auth_ticket</ticket>
<apptoken>app_token</apptoken>
 <field fid="8">party at Lindisfarne</field>
 <field fid="9">dress in style of the epoch</field>
<field fid="10">lindisfarne island</field>
<field fid="11">06-08-0793</field>
<field fid="12">10-14-1066</field>
</qdbapi>
```

Example with a truncated base64-encoded file attachment

The following request contains a truncated file in base64 encoding. Because the file is truncated, this sample won't work as shown. Replace it with your own encoded file.

```
<qdbapi>
<ticket>auth_ticket</ticket>
<apptoken>app_token</apptoken>
<udata>mydata</udata>
<field name="email">cucamonga@chuck.com</field>
<field name="assigned_number">291</field>
<field name="text">OK Corral</field>
<field name="telephone">650-345-8768.3456</field>
<field fid="22" filename="Model_T.jpg">
8D6AAAOEJJTQQIAAAAAAAQAAAAAQAAAkAAAAJAAAAAADhCSU0EBgAAAAAABwAEAAAAAQEA
4AJ0ZpbGUgd3JpdHRlbiBieSBBZG9iZSBQaG90b3Nob3CoIDQuMAD/7gAOQWRvYmUAZAAAAAAB
9sAhAAGBAQEBQQGBQUGCQYFBgkLCAYGCAsMCgoLCgoMEAwMDAwMDBAMDAwMDAwMDAwMDAwMDAwM
DAwMDAwMDAwMDAwMAQcHBw0MDRgQEBgUDg4OFBQODg4OFBEMDAwMDBERDAwMDAwMEQwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAz/wAARCAEsAWQDAREAAhEBAxEB/90ABAAt/
```

```
8QBogAAAAcBAQEBAQAAAAAAAAAABAUDAgYBAAcICQoLAQACAgMBAQEBAQAAAAAAAAABAAIDBAUG
BwgJCgsQAAIBAwMCBAIGBwMEAgYCcwECAxEEAAUhEjFBUQYTYSJxgRQykaEHFbFCI8FS0eEzFmL
wJHKC8SVDNFOSorJjc8I1RCeTo7M2F1RkdMPS4ggmgwkKGBmElEVGpLRW01UoGvLj88TU5PRldY
WVpbXF1eX1ZnaGlqa2xtbm9jdHV2d3h5ent8fX5/
c4SFhoeIiYqLjI2Oj4KTlJWWl5iZmpucnZ6fkqOkpaanqKmqq6ytrq+hEAAgIBAgMFBQQFBgQIA
wNtAQACEQMEIRIxQQVRE2EiBnGBkTKhsfAUwdHhI0IVUmJy8TMkNEOCFpJTJaJjssIHc9I14kSD
F1STCAkKGBkmNkUaJ2R0VTfyo7PDKCnT4/
OElKS0xNTk9GV1hZWltcXV5fVGVmZ2hpamtsbW5vZHV2d3h5ent8fX5/
c4SFhoeIiYqLjI2Oj4OUlZaXmJmam5ydnp+So6SlpqeoqaqrrK2ur6/
9oADAMBAAIRAxEAPwCK6Bd3Ea3CaXJDbxS2wLSSVZGVNvTmj+NTGSGIdVk+1yzFxkg0WmpdEHpc
k9vO2n21xBP+huF1ajklHWcEyxFmQer6a+or8x8CumWSx0b7mVSTO2S3mRoNOkMWnXUTGPkP3QV
2EkXpcOfAMF/Y/axsmVDqxromtzb2/
r2b+qpjjAgmSTkSY0XkpJ2J+Mslar8OZHhjkvAsv3sLp4UmuYxMPVS0ujGpkjNAKcWIDpKvwy/
Bz4rz+3+8wyxBmAk1xJDJr2l2c6xy2mlwT31xAw5Rx8jxjJ/
jXJbKLRGnfbm6foz1Dy6cPV9RvV+38Xp8vtV/3bz4YQjdZ8X1j9r67/svXp6n/UP6X/
PH1P8AizAl/9kA
</field>
```

URL alternative

https://*target_domain*/db/*target_dbid*?a=API_AddRecord&_fnm_second_year=1776
&_fid_8=changed&ticket=*auth_ticket*&apptoken=*app_token*

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_AddRecord</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>mydata</udata>
   <rid>21</rid>
   <update_id>1206177014451</update_id>
</qdbapi>
```

Did this help you? Give us a rating:

API_AddReplaceDBPage

Overview

Use API_AddReplaceDBPage to add a new database page or replace an existing page with a new page. You invoke this call on an application (dbid).

Quick Base allows you to store various types of pages at the application level. These pages can be various text or rich text or HTML page that you store and link to buttons in the Quick Base UI. They can also be XSL templates used for customizing the Quick Base application. They can also be Exact Forms, which are forms (form letters, invoices, etc.) created in Microsoft Word using the Quick Base Exact Form template that gets data from Quick Base tables.

See the Quick Base online help for more information about adding pages.

Request parameters

PARAMETER	VALUE	REQUIRED?
pagename (if adding) or pageid (if replacing)	Allows you to specify a new page to add or an existing page to replace. If you are replacing a database page, specify its page ID here. Quick Base replaces that page with the page you supply in <pagebody>. If you are adding a new page, specify the pagename.	yes
pagetype	Specifies the type of page. Valid values are: 1 – for XSL stylesheets or HTML pages. 3 – for Exact Forms.	yes
pagebody	Contains the contents of the page you are adding. Wrap this parameter in a CDATA block, as shown in the sample XML request below.	yes

PARAMETER	VALUE	REQUIRED?
ticket	Supply a valid ticket. You obtain a valid ticket from API_Authenticate. A ticket is also returned in the various API responses	yes, one of: • ticket • username/password • user token
usertoken	A user token – this is a ticket alternative, used for API access	yes, one of: • ticket • username/password • user token
apptoken	Supply a valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response. You might use this parameter to maintain state in the calling application.	no

Response values

The response to this call contains the following:

ELEMENT NAME	VALUE
action	The originating request, for example, API_AddRecord.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.

ELEMENT NAME	VALUE
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
pageID	The ID of the page that was added

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length: 490
QUICKBASE-ACTION: API_AddReplaceDBPage

<qdbapi>
<pagename>newstylesheet.xsl</pagename>
<pagetype>1</pagetype>
 <pagebody><![CDATA[
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
 <xsl:template match="/">
   <html>
    <head>
    </head>
    <body>
      Hello World
    </body>
   </html>
 </xsl:template>
</xsl:stylesheet>
]]></pagebody>
<ticket>auth_ticket</ticket>
<apptoken>app_token</apptoken>
</qdbapi>
```

URL alternative

Because URLs can only be about one thousand characters in length, this is not the recommended way to make this API call.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_AddReplaceDBPage</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <pageID>6</pageID>
</qdbapi>
```

Did this help you? Give us a rating:

API_AddSubGroup

Overview

Use API_AddSubGroup to embed a group into another group.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the group which will contain the subgroup.	yes
subgroupid	The id of the group to be embedded in the target group.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_AddSubGroup.

ELEMENT NAME	VALUE
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_AddSubGroup

 <qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <gid>345889.sdfs</gid>
   <subgroupid>820935.ksjf</subgroupid>
</qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/main?a=API_AddSubGroup&gid=345889.sdfs&subgroupid=820935.k
sjf&ticket=auth_ticket&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_AddSubGroup</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_AddUserToGroup

Overview

Use API_AddUserToGroup to add a user to a group. The user can be added as a member of the group, as a member and a manager of the group, or as a manager but not a member of the group.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the group to manage.	yes
uid	The id of the Quick Base user to add to the group.	yes
email	The email of the Quick Base user	no
screenName	The screen name of the Quick Base user, if set	no
allowAdminAccess	Whether the user should be added as an admin of the group. Default is FALSE.	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_AddUserToGroup.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_AddUserToGroup

 <qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <gid>345889.sdfd</gid>
   <uid>898790.qntp</uid>
   <allowAdminAccess>true</member>
</qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/main?a=API_AddUserToGroup&gid=345889.sdfd&uid=898790.qntp&
allowAdminAccess=true&ticket=auth_ticket&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_AddUserToGroup</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_AddUserToRole

- Overview
- Example XML Request
- Request parameters
- Response values
- Sample XML Request
- URL alternative
- Sample response

Overview

Use API_AddUserToRole to assign a user to a role in your Quick Base application.

You assign a user to a role using API_AddUserToRole. After you assign the role, you can use API_SendInvitation to invite the user to the application.

You can invoke this call on a single user several times, applying a different role each time, if you want give the user more than one role. Giving a user more than one role allows the user to sign in to the same Quick Base application with different roles. (Note that if you try to assign a role the user already has, you'll see an error message.)

Quick Base supplies standard default roles for each application: viewer, participant, and administrator. You can change these and even create custom roles for your applications. You can find out which roles are being used for a particular application using the API_GetRoleInfo call.

Note: You can also assign roles to users when you provision users. See API_ProvisionUser for more information.

To assign roles to users using the Quick Base API or UI, you must have either Basic Access with Sharing access or Full Administration access to the application. Note that, if you have Basic Access with Sharing permissions, you cannot add a user to a role that has Full Administration permissions.

Request parameters

PARAMETER	VALUE	REQUIRED?
userid	The userid of the user to be added to the access role.	yes
roleid	The ID of the access role being assigned to the user.	yes

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	Supply a valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

The response to this call contains the following:

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_AddField.
errcode	If successful, this will be 0. If there is a failure, there will be a code returned here. See Appendix A for a list of codes.
errtext	If successful, this will be "no error". If there is a failure this contains more detail on the nature of the failure.

ELEMENT NAME	VALUE
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid?
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION:API_AddUserToRole

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <userid>112245.efy7</userid>
   <roleid>10</roleid>
   <udata>whatever</udata> <!--optional pass-thru data -->
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_AddUserToRole&userid=112245.
efy7&roleid=10&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_AddUserToRole</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>whatever</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_Authenticate

Overview

Use API_Authenticate to validate a user's username and password. You invoke this call on /db/main (no dbid).

API_Authenticate validates the supplied user name and password, and, if successful, returns a ticket that is supplied in subsequent API calls. The ticket is valid for 12 hours, unless you specify a different value in the **hours** parameter. In addition to the ticket, a cookie is also returned with the name TICKET. This is the only API call that returns a ticket cookie.

Note: It is possible to use a ticket to obtain a ticket (that is, pass in a ticket instead of a username and password).

Using API_Authenticate is the equivalent of logging into Quick Base. Remember that merely logging into Quick Base by itself does not give you access rights to even a single Quick Base application. You must first be assigned a role in the application by someone who has administrator rights. (See API Overview for more information about how to use the authentication ticket when making API calls.)

You should always use secure HTTPS (not HTTP) when calling API_Authenticate. Quick Base does not support API calls over HTTP.

Note: API_Authenticate calls have a maximum time limit of 4,380 hours, or approximately 6 months. This applies to new tickets initiated on or after January 21, 2018.

Request parameters

PARAMETER	VALUE	REQUIRED?
username	The user’s email address registered with Quick Base or the user name that was selected within Quick Base when the user registered.	yes
password	The user’s Quick Base password.	yes

PARAMETER	VALUE	REQUIRED?
hours	The number of hours that the ticket will be valid. This parameter is optional. If you don't supply this parameter, the ticket will expire after 12 hours. There is a maximum time limit of 4,380 hours or approximately 6 months.	no
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_Authenticate
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
ticket	The authentication ticket needed for application access.
userid	The ID of the current user. The user ID is required for some API calls. The userID persists from session to session.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_Authenticate

<qdbapi>
   <username>PTBarnum</username>
   <password>TopSecret</password>
   <hours>24</hours>
   <udata>optional data</udata>
</qdbapi>
```

URL alternative

```
https://target_domain/db/main?a=API_Authenticate&username=PTBarnum
&password=TopSecret&hours=24
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
<action>api_authenticate</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<ticket>2_beeinrxmv_dpvx_b_crf8ttndjwyf9bui94rhciirqcs</ticket>
<userid>112245.efy7</userid>
</qdbapi>
```

Did this help you? Give us a rating:

API_ChangeGroupInfo

- Overview
- Request parameters
- Response values
- Sample XML Request
- Sample HTTP Get Request
- Sample response

Overview

Use API_ChangeGroupInfo to modify the metadata for a group. One or more elements can be modified.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the group to manage.	yes
name	The name of the new group. It may not contain spaces or punctuation.	no
description	A text description of the group.	no
accountId	The account that will own the group.	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_ChangeGroupInfo.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_ChangeGroupInfo

<qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <gid>345889.sdjl</gid>
   <name>AcmeSalesTeamLeads</name>
   <description>Team Leaders for the Acme division</description>
</qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/main?a=API_ChangeGroupInfo&gid=345889.sdjl
&name=AcmeSalesTeamLeads&Team%20Leaders%20for%20the%20Acme%20division&
ticket=auth_ticket&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_ChangeGroupInfo</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_ChangeManager

Overview

Use API_ChangeManager to assign a user to manage an app.

Request parameters

PARAMETER	VALUE	REQUIRED?
newmgr	A valid email address for the new app manager	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_ChangeManager.

ELEMENT NAME	VALUE
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_app HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_ChangeManager

<qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <newmgr>angela_leon@gmail.com</newmgr>
</qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/target_app?a=API_ChangeManager&newmgr=angela_leon@gmail.co
m
&ticket=auth_ticket&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_ChangeManager</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_ChangeRecordOwner

Overview

Use API_ChangeRecordOwner to change the owner of a record.

In Quick Base, a person who creates a record is the record owner. You can set up roles that restrict a record's view/modify access to the record owner. On occasion, you may need to transfer record ownership from one user to another. If you are using a technical support application, for instance, you might want to transfer ownership of a support incident (record) from one tech support rep to another.

You must have Full Administration rights on the application to use this call.

Request parameters

PARAMETER	VALUE	REQUIRED?
rid *or* key	The record ID. Every record in every table has a unique rid. Normally, the record ID is the key field for the record. If you set your record's key field to a field OTHER than the record id (rid), you should use the key parameter rather than rid.	yes
newowner	Specifies the user to whom you are transferring ownership. You can specify either of the following: • the user's Quick Base user name • the user's email address	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	A user token – this is a ticket alternative, used for API access	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if required by the application
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_SetRecordOwner.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_ChangeRecordOwner

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <rid>3</rid>
   <newowner>Muggsy</newowner>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_ChangeRecordOwner&rid=3
&newowner=Muggsy&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
<action>api_changerecordowner</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>mydata</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_ChangeUserRole

Overview

Use API_ChangeUserRole to assign a user to a new role, to upgrade or downgrade access, or simply to move the user to a different role.

You can also use this call to assign the user to a special role called None (role id 9) which removes the user's access to the application. Use the None role if you want to disable access while retaining the user on the application user list.

To assign roles to users using the Quick Base API or UI, you must have either Basic Access with Sharing access or Full Administration access to the application. Note that, if you have Basic Access with Sharing permissions, you cannot add a user to a role that has Full Administration permissions.

Request parameters

PARAMETER	VALUE	REQUIRED?
userid	The user ID. You can use API_GetUserInfo to get the user ID for a particular user.	yes
roleid	The user’s current role in the application	yes
newroleid	If this parameter is not supplied, or is supplied but left blank, the role id is set to 9 (None). If this parameter is supplied with an invalid role id (including 0), the API call will return an error. **Note:** If you change one user role to 9, and the user has another role in this app, the other role is still active. The user will still have the access given by that other role.	no

PARAMETER	VALUE	REQUIRED?
ticket	a ticket - needed for application access.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	Supply a valid application token.	yes, if app tokens are being used by this application.
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response. One possible use is to maintain state in the calling application.	no

Response values

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_AddField.
errcode	If successful, this will be 0. If there is a failure, there will be a code returned here. See Appendix A for a list of codes.
errtext	If successful, this will be “No error”. If there is a failure this contains more detail on the nature of the failure.

ELEMENT NAME	VALUE
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_ChangeUserRole

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <userid>112248.5nzg</userid>
   <roleid>11</roleid>
   <newRoleid/>
 </qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_ChangeUserRole
&ticket=auth_ticket&apptoken=app_token&userid=112245.efy7
&roleid=10&newRoleID=11
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>api_changeuserrole</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
</qdbapi>
```

Did this help you? Give us a rating:

API_CloneDatabase

- Overview
- Request parameters
- Response values
- Sample XML Request
- URL alternative
- Sample response

Overview

Use API_CloneDatabase to make a copy of a Quick Base application. You can choose to use the **keepData** parameter if you want to copy the application's data in addition to its structure. If you want to copy data, but exclude file attachments, you can use the **excludefiles** parameter in your request. After you copy an application, changes to the source application will not affect the copy you created, and vice versa.

When you copy a Quick Base application, all users with access to the original application automatically have access to the application copy. You can choose to remove some or all of these users from the application copy using the API_RemoveUserFromRole.

Request parameters

PARAMETER	VALUE	REQUIRED?
newdbname	Specifies a name for the new application.	yes
newdbdesc	Specifies the description for the new application.	no
keepData	Set this parameter to 1 if you want to copy the application's data. Omit this parameter if you want to copy the application structure only.	no

PARAMETER	VALUE	REQUIRED?
excludefiles	Specifies that you do not want to copy file attachments when you copy an application. If you want to copy data, but exclude file attachments from your copy: • Set the keepData parameter to 1 and • Set the excludefiles parameter to 1 If you want to copy data, including file attachments: • Set the keepData parameter to 1 and • Omit the excludefiles parameter	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	Supply a valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no
usersandroles	Specify how users and roles are treated when copying the application. Set this parameter to 1 to copy users in their roles. Set to 0 to copy users and roles separately (that is, don't assign roles). Defaults to 0.	no

Response values

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_CloneDatabase.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
newdbid	The database ID of the new database.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_CloneDatabase

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <udata>mydata</udata>
   <newdbname>YellowDots</newdbname>
   <newdbdesc>Database copy with no data</newdbdesc>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_CloneDatabase
&ticket=auth_ticket&apptoken=app_token
&keepData=1&newdbname=YellowDots&newdbdesc="Database copy with data"
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_CloneDatabase</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>mydata</udata>
   <newdbid>bddnc6pn7</newdbid>
</qdbapi>
```

Did this help you? Give us a rating:

API_CopyGroup

- Overview
- Request parameters
- Response values
- Sample XML Request
- Sample HTTP Get Request
- Sample response

Overview

Use API_CopyGroup to duplicate an existing group with the same list of users and subgroups, and a different name and ID.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the group to manage.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes
udata	Optional: Contains any udata value supplied in the request.	no
name	The name of the new group.	yes
description	A description of the group. Default is the description of the source group.	no
gacct	The account used for the group. Default is the account for the source group.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_CopyGroup.
description	The description of the new group
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
gid	The id of the new group.
name	The name of the new group.
ManagedbyUser	In this case, whether the user making the request manages the group.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_CopyGroup

<qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <gid>1213.dsfj</gid>
   <name>SalesTeamLeadsCopy</name>
   <description>Copy of the current Sales Team Leads Group</description>
</qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/main?a=API_CopyGroup&gid=1213.dsfj&name=SalesTeamLeadsCopy
&description=Copy%20ofthe%20current%20Sales%20Team%20Leads%20Group&ticket=auth_ticket
&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<qdbapi>
<action>API_CopyGroup</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<group id="1219.d47h">
<name>TestGroup</name>
<description>Copy of the current Sales Team Leads Group</description>
<managedByUser>true</managedByUser>
</group>
</qdbapi>
```

Did this help you? Give us a rating:

API_CopyMasterDetail

- Overview
- Request parameters
- Response values
- Sample XML Request
- URL alternative
- Sample response

Overview

Use API_CopyMasterDetail to copy a master record with its detail records, or to import only the detail records from one master record into a different master record. For example, you could copy a master Project record and its detail Task records. You also could import only the Task records from a selected master Project record into a different Project record.

You invoke this call on a table-level dbid for the master table to which you are copying or importing records.

API_CopyMasterDetail copies or imports all fields in selected records, regardless of the fields' permission and Auto-Fill settings. (When Auto-Fill is enabled, a field's value is copied when a user clicks the "Add Similar" button to add a new Project.)

This API does not copy file attachments with the records.

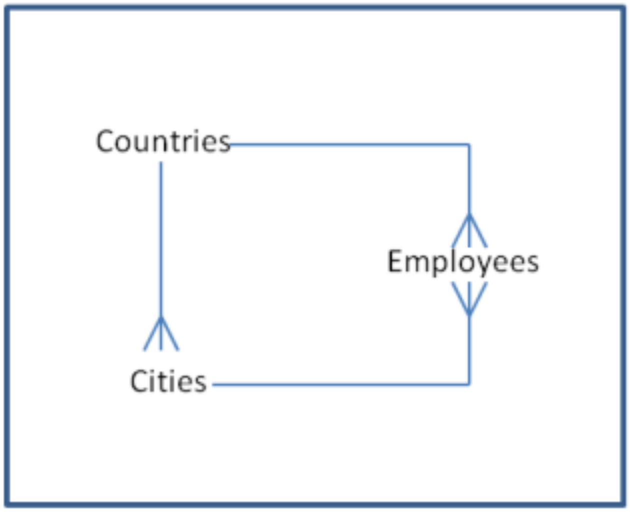
To copy or import records:

- You must have appropriate permissions: view and create permission in both the master and detail tables to copy records, and view and create permission in the detail tables to import records.
- The master table must contain a report link field as part of the relationship with each detail table. If the report link field does not exist for a relationship with a detail table, detail records for that table are not copied. You must have permission to view the report link fields.
  To confirm that the report link field exists, view the relationship and check that the master table has a report link field whose target field is the reference field listed under the detail table.
- The key field for each table from which records will be copied or imported must be Record ID.

API_CopyMasterDetail will not copy or import any records if it detects any of the following conditions:

- Records containing unique fields, such as custom key fields.
- More than ten levels of recursion
- Looping relationships. Loops occur when tables have reciprocal relationships. Because conditional reference fields can be configured with looping relationships, the API will not copy detail records from any relationship where the report link field points to a conditional reference field. (Read more about conditional reference fields.)

The figure below illustrates a looping relationship; the application contains three tables (Countries, Employees, and Cities), and has three one-to-many relationships linking each table to the other two tables:



If you have unique fields, looping relationships, or more than ten levels of recursion, you can use the **relfids** parameter to include only specific relationships, or disable recursion (by setting the **recurse** parameter to false).

How Quick Base maps predecessor relationships when it copies records

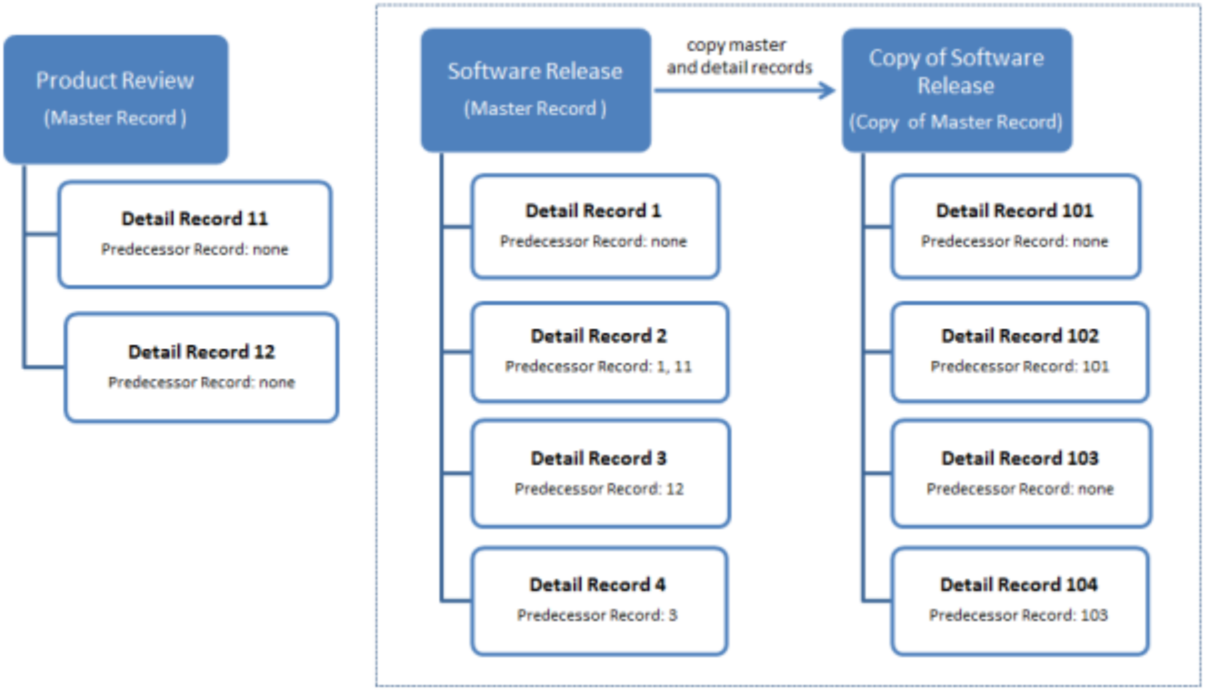
Tasks in a project usually occur in a specific order and one task cannot start until another task finishes. This means that one task *depends* on another one. A predecessor *task* is a task that must finish before another task can begin. When you specify a predecessor for a task, you set up a dependency between two tasks (which is why predecessors are sometimes also called *dependencies*). A *successor task* is a task that cannot start until another task finishes. (Read more about predecessors.)

If a detail record contains a predecessor field, Quick Base attempts to map predecessors as follows:

- If the original detail records map to each other as predecessors, Quick Base duplicates this predecessor relationship when the records are copied.
- If the original detail records map to predecessors that belong to a different master record, Quick Base cannot map the predecessor relationship because there is no corresponding copy of those detail records.
- If the original detail record maps to predecessor records that fall into both of these categories, Quick Base maps those predecessors that it can.

Consider this example of copying master and detail records in a Project Management application, illustrated below. You copy a master Project record, called "Software Release", with detail Task records 1, 2, 3, 4. The new master record is called "Copy of Software Release" and has detail records 101, 102, 103, 104. The application has another master Project record, called "Product Review", with detail Task records 11 and 12. Quick Base maps predecessor relationships as follows:

- 2 has predecessor 1 and 11. When the records are copied, Quick Base sets the predecessor field for record 102 to 101, and ignores predecessor 11.
- 3 has predecessor 12. When the records are copied, Quick Base leaves the predecessor field for record 103 blank.
- 4 has predecessor 3. When the records are copied, Quick Base sets the predecessor field for record 104 to 103.



Request parameters

PARAMETER	VALUE	REQUIRED FOR COPY (DESTRID=0)?	REQUIRED FOR IMPORT?
destrid	The record id of the destination record to which you want the records copied. If you want to copy the source master record and its detail records, set this to 0. If you want to import detail records from a different master record, set this to the record id of the master record to which you want to import the detail records.	yes	yes
sourcerid	The record id of the source record from which you want to copy detail records.	yes	yes

PARAMETER	VALUE	REQUIRED FOR COPY (DESTRID=0)?	REQUIRED FOR IMPORT?
copyfid	The field id of a text field used in the name of the new record, if destrid = 0. The name becomes “Copy of *field value".* This field cannot be a lookup, append-only text, or formula field. It also cannot be a unique or snapshot field.	yes	no
recurse	Set this parameter to true to copy all detail records associated with the master record's detail records recursively. This API supports up to 10 levels of recursion. Defaults to true.	no	no
relfids	A list of comma-separated report link field ids that specify the relationships you want to be copied. For example: `<relfids>7,9</relfids>` To copy all relationships (default), omit this parameter or set to "all".	no	no
ticket	The ticket needed for application access.	yes, one of: • ticket • username/password • user token	yes
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token	

PARAMETER	VALUE	REQUIRED FOR COPY (DESTRID=0)?	REQUIRED FOR IMPORT?
apptoken	Supply a valid application token.	yes, if app tokens are being used by this application	yes, if app tokens are being used by this application
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response. One possible use is to maintain state in the calling application.	no	no

Response values

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_CopyMasterDetail.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
parentrid	The record id of the record to which the details records were copied. This is either the destination rid that was passed in or the rid of the new master record that was created.
numcreated	The total number of new records created by the copy.

Sample XML Request

Copy master and detail records

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_CopyMasterDetail

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <udata>mydata</udata>
   <destrid>0</destrid>
   <sourcerid>1</sourcerid>
   <copyfid>6</copyfid>
</qdbapi>
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Import detail records to a different master record

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_CopyMasterDetail

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <udata>mydata</udata>
   <destrid>3</destrid>
   <sourcerid>1</sourcerid>
</qdbapi>
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

URL alternative

Copy master and detail records

```
https://target_domain/db/target_dbid?a=API_CopyMasterDetail&destRID=0&sourcerid=1
&copyfid=6&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Import detail records to a different master record

```
https://target_domain/db/target_dbid?a=API_CopyMasterDetail&destrid=3&sourcerid=1
&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_CopyMasterDetail</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <parentrid>3</parentrid>
   <numcreated>4</numcreated>
</qdbapi>
```

Did this help you? Give us a rating:

API_CreateDatabase

Overview

Use API_CreateDatabase to create a new Quick Base application. Quick Base creates the main application table populated only with built-in fields and returns an application token for your API application. After you create the application, you'll need to:

- add fields using API_AddField
- set field properties using API_SetFieldProperties
- add more tables using API_CreateTable

Quick Base returns the application-level dbid (appdbid) in the response to API_CreateDatabase. Some API calls require the use of the application-level dbid, and some require a table-level dbid. Calls requiring application-level dbids are application-wide calls; calls requiring table-level dbids are those that manipulate tables. If you are going to manipulate the table or application after creating it, you will find it useful to extract and save these values from the response values.

Request parameters

PARAMETER	VALUE	REQUIRED?
dbname	The name of the new application	yes
dbdesc	The description for the new application.	yes
ticket	The ticket needed for application access.	yes
createapptoken	Set this parameter to 1 to generate an application token for your applications. App tokens can also be generated from within Quick Base, but this method is provided for your convenience.	no

PARAMETER	VALUE	REQUIRED?
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_CreateDatabase.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Contains any udata value supplied in the request.
dbid	The table-level dbid of the new application's default table.
appdbid	The application-level dbid of the new application.
apptoken	An application token for your application to use in subsequent interactions with the Quick Base you just created. Returned only if you specified createapptoken with a value of 1 in the request.

Header and XML

```
POST https://target_domain/db/main HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_CreateDatabase
```

Sample XML Request

```
<qdbapi>
   <dbname>FuelCharter</dbname>
   <dbdesc>Vehicle and Fuel Cost Tracker</dbdesc>
   <ticket>auth_ticket</ticket>
   <createapptoken>1</createapptoken>
</qdbapi>
```

URL alternative

```
https://target_domain/db/main?a=API_CreateDatabase&ticket=auth_ticket
&dbname=FuelCharter&dbdesc=Vehicle and Fuel Cost Tracker&createapptoken=1
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_CreateDatabase</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <dbid>bddnn3uz9</dbid>
   <appdbid>bddnn3ub7</appdbid>
   <apptoken>cmzaaz3dgdmmwwksdb7zcd7a9wg</apptoken>
</qdbapi>
```

Did this help you? Give us a rating:

API_CreateGroup

- Overview
- Request parameters
- Response values
- Sample XML Request
- Sample HTTP Get Request
- Sample response

Overview

Use API_CreateGroup to create a new group. The group will be created with the caller as the group owner and the caller will also be the first user in the group.

The call needs an account id which will be the associated account for the group, unless the call is made to a realm with only one account. If the realm only has one account, then the account associated with the realm will be the group account.

If a realm has more than one account, the **accountID** parameter will be used to disambiguate the associated account for the group.

Permissions

The caller (user) must be the manager of the account where the group is created.

Request parameters

PARAMETER	VALUE	REQUIRED?
name	The name of the new group. It may not contain spaces or punctuation.	yes
description	A text description of the group.	yes
accountId	Optional. The account that will own the group.	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes

PARAMETER	VALUE	REQUIRED?
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_CreateGroup.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
group	The XML structure of the group.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_CreateGroup

<qdbapi>
   <udata>misc data</udata>
   <name>MarketingSupport</name>
   <description>Support staff for sr marketing group</description>
   <accountID>456789</accountID>
   <ticket>auth_ticket</ticket>
</qdbapi>
```

Sample XML Get Request

```
https://target_domain/db/main?a=API_CreateGroup&accountID=212968
&name=MarketingSupport&description=Support%20staff%20for%20sr%20marketing%20grou
p
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<qdbapi>
<action>API_CreateGroup</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<group id="1217.dgpt">
<name>MarketingSupport</name>
<description>Support staff for sr marketing group</description>
<managedByUser>true</managedByUser>
</group>
</qdbapi>
```

Did this help you? Give us a rating:

API_CreateTable

Overview

Use API_CreateTable to create a table for an application. You must have application administration rights to use this call. The dbid you supply must be an application-level dbid, not a table-level dbid.

Request parameters

PARAMETER	VALUE	REQUIRED?
tname	The name you want to use for the name of the table. If you don't supply this, the name *Records* is assigned by default. Because your application requires unique table names, you'll be able to use the default only once.	no, but recommended
pnoun	The name you want to use for records in the table. You should use a plural noun that can apply to all records in the table. For example, if each table row contains information about a vehicle that your company owns, you might want to use "Vehicles" as your pnoun. If you don't supply this, the name *Records* is assigned by default.	no, but recommended
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: - ticket - username/password - user token
apptoken	Supply a valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_CreateTable.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Contains any udata value supplied in the request.
newdbid	The dbid of the new table.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_CreateTable

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <tname>My Vehicle List</tname>
   <pnoun>Vehicles</pnoun>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_CreateTable&ticket=auth_ticket
&apptoken=app_token&tname=MyVehicleList&pnoun=Vehicles
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_createtable</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <newDBID>bddfa5nbx</newDBID>
</qdbapi>
```

Did this help you? Give us a rating:

API_DeleteDatabase

Overview

Use API_DeleteDatabase to delete a table or an entire application. Supply an application-level dbid to delete an application; supply a table-level dbid to delete a table.

If you use this API to delete an app, you must be the app manager, or the API will return an error. If you use this API to delete a table, you must have Full Administration permissions in the app, or the API will return an error.

Request parameters

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_DeleteDatabase.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_DeleteDatabase

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_DeleteDatabase
&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_DeleteDatabase</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
</qdbapi>
```

Did this help you? Give us a rating:

API_DeleteField

Overview

Use API_DeleteField to delete a field by specifying the field id (fid). You invoke this call on a table-level dbid. (If you use an application level dbid, Quick Base returns an error.)

You must have Full Administration permissions on the application to use this call.

This call deletes the field for all records, regardless of whether the field contains data. Note that the call does not issue a warning message before the deletion.

Request parameters

PARAMETER	VALUE	REQUIRED?
fid	The field ID of the field to be deleted.	yes
ticket	a ticket – needed for application access	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	Supply a valid application token.	yes, if app tokens are being used by this application.

PARAMETER	VALUE	REQUIRED?
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response. One possible use is to maintain state in the calling application.	no

Response values

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_AddField.
errcode	If successful, this will be 0. If there is a failure, there will be a code returned here. See Appendix A for a list of codes.
errtext	If successful, this will be “No error”. If there is a failure this contains more detail on the nature of the failure.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_DeleteField

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <fid>8</fid>
   <udata>whatever</udata> <!--optional-->
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_DeleteField&fid=FID&ticket=auth_ticket
&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_DeleteField</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
</qdbapi>
```

Did this help you? Give us a rating:

API_DeleteGroup

- Overview
- Request parameters
- Response values
- Sample XML Request
- Sample HTTP Get Request
- Sample response

Overview

Use API_DeleteGroup to delete a group.

Caution: Use this call with care. Once a group has been deleted it cannot be restored.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the group to delete.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_DeleteGroup.

ELEMENT NAME	VALUE
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_DeleteGroup

<qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <gid>345889.skef</gid>
</qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/main?a=API_DeleteGroup&gid=345889.skef
&ticket=auth_ticket&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_DeleteGroup</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_DeleteRecord

Overview

Use API_DeleteRecord to delete a single record from a table. You invoke this call on a table-level dbid. If you use an application level dbid, you'll get an error.

You must have Full Administration permissions on the application to use this call.

Note that this call allows you to delete one record at a time. If you want to delete several records at once, see API_PurgeRecords.

Request parameters

PARAMETER	VALUE	REQUIRED?
rid or key	The record ID of the record to be deleted. You can obtain the record ID of any record in a query. (See API_DoQuery). If you use a key field OTHER than the record id (rid), you should use the <key> parameter rather than <rid>.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	A user token – this is a ticket alternative, used for API access	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
apptoken	Supply a valid application token.	yes, if the application requires application tokens.
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_DeleteRecord.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
rid	The record ID of the deleted record.
udata	Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_DeleteRecord
```

```
<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <rid>18</rid>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_DeleteRecord&rid=18
&udata=mydata&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_DeleteRecord</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <rid>18</rid>
</qdbapi>
```

Did this help you? Give us a rating:

API_DoQuery

Overview

Use API_DoQuery to get records from a table. You invoke this call on a table-level dbid. You can use the parameters (described below) to define which columns will be returned, how many records will be returned and how they will be sorted, and whether the Quick Base should return structured data.

When you build a query, you must build at least one query string. A query string is composed of:

- a field ID (fid)
- an uppercase comparison operator (see the table below for a list of available operators).
- the value to be compared against (the matching value)

You should separate each of these query string components using a period and enclose the entire query string in curly braces, as shown below:

```
{'fid'.operator.'matching_value'}
```

Examples:

```
<query>{'5'.CT.'Ragnar Lodbrok'}</query>
```

This example specifies that Quick Base should return all records where the field with fid *5* contains the value *Ragnar Lodbrok*.

```
<query>{7.EX.’Ragnar Lodbrok’}AND{9.CT.’Erich’}</query>
```

This example specifies that Quick Base should return all records where *both* of the following conditions are true:

- fid *7* equals *Ragnar Lodbrok*
- fid *9* contains *Erich*

```
<query>({'6'.EX.'Six'}OR{'6'.EX.'Seven'})AND{'7'.GTE.'2'}</query>
```

This example specifies that Quick Base should return all records where fid 6 is equal to *Six* or *Seven*, and all records where fid 7 is greater than or equal to *2*.

Specifying a value as the matching value

You can create queries that find records where the matching value is a certain value that you specify. To create a query that finds a specific matching value, you enter the value in single quotes in the query string.

Using the value in another field as the matching value

For most field types, you can write queries that compare one field with another field. When you write queries in this way, you set the matching value to a field rather than a specific value. For instance, in a project management application, you might want to write queries that will find any of the following results:

- All records where the projects were completed on the originally forecasted completion date. Here, your query should find all records where value in the Actual Completion date field matches exactly the value in the Projected Completed date field.
- All records where the Notes field contains the name of one of your business contacts. Here, your query should find all records where the Notes field contains the value in the Contact field.
- All records where tasks are assigned to users who are not project leads. Here, your query should finds records where the value in the Assigned To field IS NOT EQUAL TO the value in the Project Lead field

To compare the value in one field to the value in another field, your query should be made up of the original fid, the comparison operator, and the field ID to be compared against, preceded by the string _fid_ in the query string. The following table shows an example of a matching value set to a specific value, and one set to the value in another field.

MATCHING VALUE	EXAMPLES AND RESULTS

Specific value	XML Example:
	`<query>{7.EX.'Ragnar Lodbrok'}</query>`
	URL Example:
	`http://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.EX.'Ragnar Lodbrok'}`
	This query returns:
	All records where the value in fid 7 is equal to the string 'Ragnar Lodbrok'.

The value in another field	XML Example:
	`<query>{7.XEX.'_FID_10'}</query>`
	URL Example:
	`http://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.XEX.'_FID_10'}`
	This query returns:
	All records where the value in fid 7 is not equal to the value in fid 10.

Note that the user must have permission to access the field in the matching value to get expected results. In addition, the field type of the matching value field must match the field type of the field being evaluated.

Field types that can be used as a query's matching value

The following table lists the field types you can use as the matching value in a query:

FIELD TYPE	NOTES
Text	You can compare any Text field with any other Text field. Text fields are: • Text • Text - Multi-line • Text - Multiple Choice • Formula - Text Note that you can mix and match the types above--that is, you can compare a simple Text field with a Formula-Text field in your query. Example: `<query>{16.CT.’Customer’}</query>` This query returns: All records where the value in fid 16 contains customer.
Multi-select Text	You can compare Multi-select Text fields with other Multi-select Text fields. Example: `<query>{90.HAS.’Framingham;Walpole’}</query>` This query returns: All records where the value in fid 90 contains Framingham and Walpole.

FIELD TYPE	NOTES
Numeric	You can compare any Numeric field with any other Numeric field. Numeric fields are: • Numeric • Numeric - Currency • Numeric - Percent • Numeric - Rating • Formula - Numeric Note that you can mix and match the types above--that is, you can compare a simple Numeric field with a Numeric - Percent field in your query. Example: `<query>{72.LT.150}</query>` This query returns: All records where the value in fid 72 is less than 150.

FIELD TYPE	NOTES

Date	You can compare any Date field with any other Date field. Date fields are:
	Date
	Date/Time
	Formula - Date/Time
	Note that you can mix and match the types above--that is, you can compare a Date field with a Date/Time field.
	Example (Date):
	`<query>{45.BF.’01-01-2016’}</query>`
	This query returns:
	All records where the value in fid 45 is before 01-01-2016.
	Example (Date/Time):
	`<query>{2.BF.’11-01-2016 8:00AM’}</query>`
	This query returns:
	All records where the value in fid 2 is before 11-01-2016 at 8:00 AM.

Time of Day	You can compare Time of Day fields with other Time of Day fields or Formula - Time of Day fields. Example: `<query>{2.BF.’11:00AM’}</query>` This query returns: All records where the value in fid 2 is before 11:00 AM.

FIELD TYPE	NOTES
Checkbox	You can compare Checkbox fields with other Checkbox fields or Formula - Checkbox fields. Example: `<query>{86.EX.’true’}</query>` This query returns: All records where the value in fid 86 is true.
Phone Number	You can compare Phone Number fields with other Phone Number fields or Formula - Phone Number fields. Example: `<query>{87.EX.’(617) 250-1234’}</query>` This query returns: All records where the value in fid 87 is (617) 250-1234.
Email Address	You can compare Email Address fields with other Email Address fields or Formula - Email Address fields. Example: `<query>{77.EX.’Jeanne_Smith@acme.com’}</query>` This query returns: All records where the value in fid 77 is Jeanne_Smith@acme.com.

FIELD TYPE	NOTES
User	You can compare User fields with other User fields and Formula - User fields. Example: `<query>{4.EX.'Jeanne_Smith@acme.com'}</query>` This query returns: All records where the value in fid 4 is Jeanne_Smith@acme.com.
List-User	You can compare List-User fields with other List-User fields and Formula - List-User fields. Example: `<query>{88.HAS.'John_Rogers@acme.com;Jeanne_Smith@acme.com'}</query>` This query returns: All records where the value in fid 88 contains John_Rogers@acme.com and Jeanne_Smith@acme.com.
URL	You can compare URL fields with other URL fields and Formula - URL fields. Example: `<query>{89.EX.' https://www.quickbase.com'}</query>` This query returns: All records where the value in fid 89 is https://www.quickbase.com.

Field types that can't be used as a query's matching value

You cannot use the following field types as the matching value in a query:

- Predecessor
- iCalendar
- vCard
- File Attachment

- Report Link
- Duration

For more details about how to use these parameters, see Building and Using Queries.

Request parameters

PARAMETER	VALUE	REQUIRED?
query or qid or qname	Specifies the query. You can use any of these three options: • Use `<query>{myfid.operator.value}</query>` to build your own query string. • Use `<qid>`*n*`</qid>`if you want to use a saved query and want to use its query ID. • Use `<qname>`*myQuery*`</qname>` if you want to use a saved query and refer to it by its query name. **Note:** API_DoQuery can be used to create the query used by API_GenResultsTable.	no (returns all records if absent)
clist	A period-delimited list of field IDs to be returned. Quick Base will return values for these field IDs in the order in which you enter them here. To return all fields in a table, set this parameter to the value a. Omit this parameter if you want the query to return the table's default columns.	no

PARAMETER	VALUE	REQUIRED?

slist	A period-delimited list of field IDs used to determine sorting as follows:
	`<slist>`*fid*`</slist>`
	The following slist parameter sorts all records by the field whose fid is 7.
	`<slist>7</slist>`
	You can specify a secondary sort by including another fid in the string (separating each fid with a period)
	`<slist>7.12</slist>`
	URL example:
	`&slist=7.12`
	If you don’t specify the sort list, the default sort specified on the Report Layout page determines the order in which records are displayed.
		no

<table>
<tr><th>PARAMETER</th><th>VALUE</th><th>REQUIRED?</th></tr>
<tr><td>fmt</td><td>
<p>Set this parameter to <em>structured</em> to specify that the query should return structured data. Omit this parameter if you do not want the query to return structured data.</p>
<p>If you use this parameter:</p>
<ul>
<li>Field IDs (fids) are returned rather than field names.</li>
<li>URLs are returned for file attachments.</li>
<li>The display format field property is included with Numeric fields. The values for this property are:
<table>
<tr><th>VALUE</th><th>DISPLAY FORMAT</th></tr>
<tr><td>0</td><td>12345678.00</td></tr>
<tr><td>3</td><td>12,345,678.00</td></tr>
<tr><td>6</td><td>12.345.678,00</td></tr>
<tr><td>7</td><td>12345678,00</td></tr>
<tr><td>8</td><td>1,23,45,678.00</td></tr>
<tr><td>10</td><td>1.23.45.678,00</td></tr>
</table>
</li>
<li>The display format, currency symbol, and symbol placement field properties are included with Numeric - Currency fields.
The values for the symbol placement field property are:
<table>
<tr><th>VALUE</th><th>DISPLAY FORMAT</th></tr>
<tr><td>0</td><td>before the number ($-1.00)</td></tr>
<tr><td>1</td><td>between any negative sign and the number (-$1.00, default)</td></tr>
<tr><td>2</td><td>after the number (-1.00$)</td></tr>
</table>
</li>
</ul>
</td><td>no</td></tr>
</table>

PARAMETER	VALUE	REQUIRED?
returnpercentage	Specifies whether Numeric - Percent values in the returned data will be percentage format (10% is shown as 10) or decimal format (10% is shown as .1). Set this parameter to 0 to return decimal format (the default) or 1 to return percentage format.	no
options	Specifies return options for the query. You can use any or all of the options, separating the options with a period. num-n --specifies a maximum of n records in the return onlynew --returns only those records marked with new or updated flags skp-n --skips the first n records returned by the query sortorder-A --specifies an ascending order sortorder-D --specifies a descending order **Tip:** To sort on multiple fields, simply add more A's or D's onto the sortorder option: sortorder-ADA will sort ascending by the first field in the query, descending by the second field, and ascending by the third field. nosort --returns unsorted records, ignoring the sortorder option, the slist parameter, and the default sort for the table	no
includeRids	Specifies that the record IDs of each record should be returned. To return record IDs for each record, set this parameter to 1.	no
useFids	Set this parameter to 1 to specify that the field ids of each field should be used for the field tags in the record aggregate, instead of field names, when fmt is not specified for the request. The field tags will match those in the structured response.	no

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Query string comparison operators

Note: Query string comparison operators must be in uppercase.

COMPARISON OPERATOR	DESCRIPTION
CT	Contains either a specific value or the value in another field of the same type. (Do not use this operator with List - User and Multi-select Text fields; instead, use HAS.)

COMPARISON OPERATOR	DESCRIPTION
XCT	Does not contain either a specific value, or the value in another field of the same type. (Do not use this operator with List - User and Multi-select Text fields; instead, use XHAS.)
HAS	Used with List - User and Multi-select Text fields only. Specifies that the field contains a specific set of values. For each user you are trying to find in a List - User field, you must enter the user's ID, user name, or email address. You can also enter placeholder names. Be sure to surround placeholder names with double quotes. The query parameter must be surrounded by single quotes. Separate values in the list using a semi-colon. For example: `<query>{'6'.HAS.'-8675309; -9873297'}</query>`
XHAS	Used with List - User and Multi-select Text fields only. Specifies that the field does not contain a specific set of values. See Filtering records using fields with multiple values for more information. For each user you are trying to find in a List - User field, the query parameter must contain the user's ID, email address, or user name. You can also specify a placeholder name. Placeholder names must be enclosed in double quotes. The entire query parameter must be surrounded by single quotes. Separate values in the list using semi-colons. Note that a matching record must contain all users you specify. This query: `<query>{'6'.XHAS. '-8675309; -9873297'}</query>` ...specifies that you want to see records that do not contain BOTH of these users. Therefore, the query will return records that contain either one or neither, but not both, of these users.

COMPARISON OPERATOR	DESCRIPTION
EX	Is equal to either a specific value, or the value in another field of the same type. When specifying values to query from List - User and Multi-select Text fields, enclose the entire query parameter in single quotes. Separate the values you're looking for using semi-colons. See Filtering records using fields with multiple values for more information.
TV	True Value (compares against the underlying foreign key or record ID stored in relationship fields.) Also used for queries on user fields.
XEX	Is not equal to either a specific value, or the value in another field of the same field type. When specifying values to query from List - User and Multi-select Text fields, enclose the entire query parameter in single quotes. Separate the values you're looking for using semi-colons.
SW	Starts with either a specific value or the value in another field of the same type.
XSW	Does not start with either a specific value or the value in another field of the same type.
BF	Is before either a specific value or the value in another field of the same type.
OBF	Is on or before either a specific date or the value in another date field
AF	Is after either a specific date or the value in another date field

COMPARISON OPERATOR	DESCRIPTION
OAF	Is on or after either a specific date or the value in another date field
IR	Is during. Use this operator with date and date/time fields, to determine whether a particular date falls within particular date range relative to the current date. Learn more about relative date ranges.
XIR	Is not during. Use this operator with date and date/time fields, to determine whether a particular date does not fall within a particular date range relative to the current date. Learn more about relative date ranges.
LT	Is less than either a specific value or the value in another field of the same type.
LTE	Is less than or equal to either a specific value or the value in another field of the same type.
GT	Is greater than either a specific value or the value in another field of the same type.
GTE	Is greater than or equal to either a specific value or the value in another field of the same type.

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_DoQuery.

ELEMENT NAME	VALUE
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
records	The records returned depend on whether you specified the **fmt** parameter If you did NOT specify **fmt**, the query returns: • A dbinfo aggregate • a variables aggregate for the DBVars • a chdbids aggregate (empty, since this API is called on a table) • one record aggregate for each returned record If you DID specify **fmt**, the query returns a <table> super-aggregate containing the following aggregates: • `original` (table metadata such as next available fid, next available rid, create/mod dates, and some defaults) • `variables` (all the DBVars) • `queries` (all of the saved queries, with qid and qnames) • `fields` (properties for every field to be returned in the query) • `records` (contains 0-N <record> sub-aggregates containing the actual record data, using fids, not field names)

You can search for Dates formatted according to the app's properties, but Dates are returned as milliseconds since January 1st, 1970 00:00:00 UTC, which is the same internal representation used by JavaScript.

Numeric values are returned with no currency symbol or separator, and a period as decimal, like *12345678.00*.

Text - Multi-line fields are returned with each line of the text field separated by a `<br />` tag.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_DoQuery

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <udata>mydata</udata>
   <query>{'5'.CT.'Ragnar Lodbrok'}AND{'5'.CT.'Acquisitions'}</query>
   <includeRids>1</includeRids>
   <clist>5.6.7.22.3</clist>
   <slist>3</slist>
   <options>num-4.sortorder-A.skp-10.onlynew</options>
   <fmt>structured</fmt>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_DoQuery&includeRids=1
&ticket=auth_ticket&apptoken=app_token&udata=mydata
&query={'5'.CT.'Ragnar Lodbrok'}AND{'5'.CT.'Acquisitions'}&clist=5.6.7.22.3
&slist=3&options=num-4.nosort.skp-10.onlynew&fmt=structured
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*.
Read about this notation.

Sample response (structured format)

```
<?xml version="1.0" ?>
<qdbapi>
<action>API_DoQuery</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>mydata</udata>
   <qid>-1</qid>
   <qname />
<table>
   <name>API created Sample</name>
   <desc>This is a sample table.</desc>
   <original>
      <table_id>bh9ckdaue</table_id>
      <app_id>bh9ckc9ft</app_id>
      <cre_date>1204586581894</cre_date>
      <mod_date>1206583187767</mod_date>
      <next_record_id>34</next_record_id>
      <next_field_id>24</next_field_id>
      <next_query_id>5</next_query_id>
      <def_sort_fid>6</def_sort_fid>
      <def_sort_order>1</def_sort_order>
   </original>
<variables>
   <var name="Blue">14</var>
   <var name="Jack">14</var>
```

```
      <var name="Magenta">12</var>
      <var name="usercode">14</var>
   </variables>
   <queries>
      <query id="1">
         <qyname>List All</qyname>
         <qytype>table</qytype>
         <qycalst>0.0</qycalst>
      </query>
      <query id="2">
         <qyname>List Changes</qyname>
         <qytype>table</qytype>
         <qydesc>Sorted by Date Modified</qydesc>
         <qyslst>2</qyslst>
      <qyopts>so-D.onlynew.</qyopts>
      <qycalst>0.0</qycalst>
      </query>
   </queries>
   <fields>
      <field id="5" field_type="userid" base_type="text" role="modifier">
         <label>Last Modified By</label>
         <nowrap>1</nowrap>
         <bold>0</bold>
         <required>0</required>
         <appears_by_default>0</appears_by_default>
         <find_enabled>1</find_enabled>
         <allow_new_choices>1</allow_new_choices>
         <sort_as_given>0</sort_as_given>
         <carrychoices>1</carrychoices>
         <foreignkey>0</foreignkey>
         <unique>0</unique>
         <doesdatacopy>0</doesdatacopy>
         <fieldhelp />
         <display_user>fullnamelf</display_user>
         <default_kind>none</default_kind>
      </field>
      <field id="6" field_type="phone" base_type="text">
         <label>Business Phone Number</label>
         <nowrap>0</nowrap>
         <bold>0</bold>
         <required>1</required>
         <appears_by_default>1</appears_by_default>
         <find_enabled>1</find_enabled>
         <allow_new_choices>0</allow_new_choices>
         <sort_as_given>0</sort_as_given>
         <carrychoices>0</carrychoices>
         <foreignkey>0</foreignkey>
         <unique>1</unique>
         <doesdatacopy>0</doesdatacopy>
         <fieldhelp>This is the phone number</fieldhelp>
         <num_lines>1</num_lines>
         <append_only>0</append_only>
         <allowHTML>0</allowHTML>
         <has_extension>1</has_extension>
      </field>
      <field id="7" field_type="email" base_type="text">
         <label>Email</label>
         <nowrap>0</nowrap>
         <bold>0</bold>
         <required>0</required>
         <appears_by_default>1</appears_by_default>
         <find_enabled>1</find_enabled>
         <allow_new_choices>0</allow_new_choices>
         <sort_as_given>0</sort_as_given>
         <carrychoices>1</carrychoices>
         <foreignkey>0</foreignkey>
         <unique>0</unique>
         <doesdatacopy>1</doesdatacopy>
         <fieldhelp />
      </field>
      <field id="22" field_type="file" base_type="text">
         <label>File</label>
         <nowrap>0</nowrap>
         <bold>0</bold>
         <required>0</required>
         <appears_by_default>1</appears_by_default>
         <find_enabled>1</find_enabled>
         <allow_new_choices>0</allow_new_choices>
         <sort_as_given>0</sort_as_given>
         <carrychoices>1</carrychoices>
         <foreignkey>0</foreignkey>
         <unique>0</unique>
         <doesdatacopy>0</doesdatacopy>
         <fieldhelp />
         <max_versions>3</max_versions>
         <see_versions>1</see_versions>
         <use_new_window>1</use_new_window>
      </field>
      <field id="3" field_type="recordid" base_type="int32" role="recordid"
   mode="virtual">
         <label>Record ID</label>
         <nowrap>1</nowrap>
         <bold>1</bold>
         <required>0</required>
         <appears_by_default>0</appears_by_default>
         <find_enabled>1</find_enabled>
         <allow_new_choices>0</allow_new_choices>
         <sort_as_given>0</sort_as_given>
         <default_value>10</default_value>
         <carrychoices>1</carrychoices>
         <foreignkey>0</foreignkey>
         <unique>1</unique>
         <doesdatacopy>0</doesdatacopy>
         <fieldhelp />
         <comma_start>0</comma_start>
         <does_average>0</does_average>
         <does_total>0</does_total>
         <blank_is_zero>0</blank_is_zero>
      </field>
   </fields>
   <lastluserid>0</lastluserid>
   <lusers>
      <luser id="112149.bhsv">AppBoss</luser>
   </lusers>
```

```
<records>
   <record>
      <f id="5">112149.bhsv</f>
      <f id="6">(123) 333-4321 x34566</f>
      <f id="7">bo@co.com</f>
      <f id="22" />
      <f id="3">12</f>
      <update_id>1206421031556</update_id>
   </record>
   .
   .
   .
</records>
</table>
</qdbapi>
```

Did this help you? Give us a rating:

Understanding relative date ranges

You can write Quick Base queries to find records where certain dates fall within relative date ranges. A relative date range is a period of time that is relative to the current date. (Last Week, Next Month, and During the Next 3 Quarters are examples of relative date ranges.

Using relative date ranges allows you to create reports that you can run periodically without needing to alter your query. For example, you may want to create a report that you run monthly, that shows all tasks with completion dates that occurred during the last month.

The IR and XIR operators

You can write a query using the IR (Is during) operator to find all records where the completion date or date/time field falls within the relative date range of Last Month.

You can also use the XIR (is not in range) operator to find records with date or date/time fields that do not fall within the specified dynamic date range.

You can use either of these operators with any date or date/time field in your applications. You can build queries that use dynamic date ranges in the following categories:

- Days
- Weeks
- Months
- Quarters
- Years

Relative date ranges: Days

A day starts at 12:00:00 a.m. and continues for 24 hours, ending at 11:59:59 p.m. (Start and end times are always based on the application's time zone.)

OPERATOR	DESCRIPTION

today	Starts at 12:00:00 a.m. on the current day and continues for 24 hours.
	XML example:
	`<query>{7.IR.'today'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'today'}`
	This example returns:
	All records where the date or date/time in fid 7 is equal to the current day.

OPERATOR	DESCRIPTION
tomorrow	Starts at 12:00:00 a.m. on the day after the current day and continues for 24 hours. XML example: `<query>{7.IR.'tomorrow'}</query>` URL example: https://*target_domain*/db/*target_dbid*?a=API_DoQuery&query={7.IR.'tomorrow'} This example returns: All records where the date or date/time in fid 7 is equal to tomorrow's date.
yesterday	Starts at 12:00:00 a.m. on the day before the current day and continues for 24 hours. XML example: `<query>{7.IR.'yesterday'}</query>` URL example: https://*target_domain*/db/*target_dbid*?a=API_DoQuery&query={7.IR.'yesterday'} This example returns: All records where the date or date/time in fid 7 is equal to yesterday's date.
next N days	Starts at 12:00:00 a.m. tomorrow and continues for the next N days. (The range does not include today.) XML example: `<query>{7.IR.'next 5 d'}</query>` URL example: https://*target_domain*/db/*target_dbid*?a=API_DoQuery&query={7.IR.'next+5+d'} This example returns: All records where the date or date/time in fid 7 falls within the 5 consecutive days starting at 12:00.00 a.m. tomorrow.

OPERATOR	DESCRIPTION

last N days	Starts at 12:00:00 a.m. N days before the current day and continues to the end of yesterday. (The range does not include the current day.)
	XML example:
	`<query>{7.IR.'last 5 d'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'last+5+d'}`
	This example returns:
	All records where the date or date/time in fid 7 falls within the 5 consecutive days that end on 11:59:59 yesterday.

Relative date ranges: Weeks

Each application manager can define the day that starts a week for each application. A week starts at 12:00:00 a.m. on the specified start date and continues 7 consecutive days, ending at 11:59:59 on the seventh day. (Start and end times are always based on the application's time zone.)

By default, a week starts at 12:00:00 a.m. on Sunday and continues for 7 complete days (ending at 11:59:59 on Saturday.)

See the Quick Base online help to learn how to set the day of the week for the application.

OPERATOR	DESCRIPTION

this wk	Starts at 12:00:00 a.m. on the first day of the current week and continues for seven days.
	XML example:
	`<query>{7.IR.'this wk'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'this+wk'}`
	This example returns:
	All records where the date in fid 7 occurs within the time period that starts at 12:00:00 a.m. on the first day of the current week and continues for seven days.

OPERATOR	DESCRIPTION

next wk	The 7-day period that starts immediately after the current week ends.
	Next week starts at 12:00:00 a.m. the first day of the week after the current week and continues for 7 days.
	XML example:
	`<query>{7.IR.'next wk'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'next+wk'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the 7-day period that starts at 12:00:00 a.m. on the first day of the week after the current week.

last wk	The 7-day period that ends immediately before the current week starts.
	Last week starts at 12:00:00 a.m. on the first day of the week that precedes the current week and continues for seven days.
	XML example:
	`<query>{7.IR.'last wk'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'last+wk'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the 7-day period that ends at 11:59:59 on the last day of the week before the current week.

OPERATOR	DESCRIPTION
next N wk	Starts at 12:00:00 a.m. on the first day of next week and continues for the next N weeks. XML example: `<query>{7.IR.'next 5 wk'}</query>` URL example: https://*target_domain*/db/*target_dbid*?a=API_DoQuery&query={7.IR.'next+5+wk'} This example returns: All records where the date or date/time in fid 7 occurs within the 5 consecutive weeks starting next week.
last N wk	Starts at 12:00:00 a.m. N weeks before the current week and continues for n weeks. XML example: `<query>{7.IR.'last 5 wk'}</query>` URL example: https://*target_domain*/db/*target_dbid*?a=API_DoQuery&query={7.IR.'last+5+wk'} This example returns: All records where the date or date/time in fid 7 falls within the 5 consecutive weeks that end immediately before the current week begins.

Dynamic date ranges: Months

In Quick Base, months are defined according to the Western calendar. A month starts at 12:00:00 a.m. on the first day of a month and continues until 11:59:59 on the last day of the month. (Start and end times are always based on the application's time zone.)

OPERATOR	DESCRIPTION

this mon	Starts at 12:00:00 a.m. on the first day of the current month and continues until 11:59:59 on the last day of the month.
	XML example:
	`<query>{7.IR.'this mon'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'this+mon'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the current month.

next mon	Starts at 12:00:00 a.m. on the first day of the month that follows the current month and continues until 11:59:59 on the last day of the month.
	XML example:
	`<query>{7.IR.'next mon'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'next+mon'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the month immediately following the current month

OPERATOR	DESCRIPTION
last mon	Starts at 12:00:00 a.m. on the first day of the month that precedes the current month and continues until 11:59:59 on the last day of the month. XML example: `<query>{7.IR.'last mon'}</query>` URL example: https://*target_domain*/db/*target_dbid*?a=API_DoQuery&query={7.IR.'last+mon'} This example returns: All records where the date or date/time in fid 7 occurs within the month immediately preceding the current month.
next N mon	Starts at 12:00:00 a.m. on first day of the next month and continues for N months. (The range does not include the current month.) XML example: `<query>{7.IR.'next 5 mon'}</query>` URL example: https://*target_domain*/db/*target_dbid*?a=API_DoQuery&query={7.IR.'next+5+mon'} This example returns: All records where the date or date/time in fid 7 occurs within the 5 consecutive months starting the on the first day of next month.

OPERATOR	DESCRIPTION

last N mon	Starts at 12:00:00 a.m. N months before the current month and continues for N months. (The range does not include the current month.)
	XML example:
	`<query>{7.IR.'last 5 mon'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'last+5+mon'}`
	This example returns:
	All records where the date or date/time in fid 7 falls within the 5 consecutive months that end immediately before the current month begins.

Relative date ranges: Quarters

Each year is divided into 3-month long quarters. In Quick Base, quarters are defined as follows:

- Q1--January through March
- Q2--April through June
- Q3--July through September
- Q4--October through December

(Start and end times are always based on the application's time zone.)

OPERATO R	DESCRIPTION

this q	Starts at 12:00:00 a.m. on the first day of the current quarter and continues for three months, ending at 11:59:59 on the last day of the quarter.
	XML example:
	`<query>{7.IR.'this q'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'this+q'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the current quarter.

next q	Starts at 12:00:00 a.m. on the first day of the quarter after the current quarter ends, and continues for three months, ending at 11:59:59 on the last day of that quarter.
	XML example:
	`<query>{7.IR.'next q'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'next+q'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the quarter following the current quarter.

OPERATOR	DESCRIPTION

last q	Starts at 12:00:00 a.m. on the first day of the quarter immediately preceding the current quarter and continues for three months, ending at 11:59:59 on the last day of the quarter.
	XML example:
	`<query>{7.IR.'last q'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'last+q'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the quarter immediately preceding the current quarter.

next N q	Starts at 12:00:00 a.m. on first day of the quarter immediately following the current quarter and continues for N quarters. (The range does not include the current quarter.)
	XML example:
	`<query>{7.IR.'next 5 q'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'next+5+q'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the 5 consecutive quarters starting the on the first day of next quarter.

OPERATOR	DESCRIPTION

last N q	Starts at 12:00:00 a.m. N quarters before the current quarter and continues for N quarters. (The range does not include the current quarter.)
	XML example:
	`<query>{7.IR.'last 5 q'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'last+5+q'}`
	This example returns:
	All records where the date or date/time in fid 7 falls within the 5 consecutive quarters that end immediately before the current quarter begins.

Relative date ranges: Fiscal Quarters

Fiscal year definitions vary from organization to organization. For each Quick Base application, you can define the fiscal year your company uses. You define the starting month of your fiscal year; the fiscal year continues for 12 months following the starting month. (Start and end times are always based on the application's time zone.)

Quick Base automatically calculates your fiscal quarters based on the fiscal year starting month.

If your application manager does not set the fiscal year, Quick Base uses the calendar year as the default.

See the Quick Base online help to learn how to define the fiscal year for the application.

OPERATOR	DESCRIPTION

this fq	Starts at 12:00:00 a.m. on the first day of the current fiscal quarter and continues for three months, ending at 11:59:59 on the last day of the third fiscal quarter.
	XML example:
	`<query>{7.IR.'this fq'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'this+fq'}`
	This example returns:
	All records where the date in fid 7 occurs within the current fiscal quarter.

next fq	Starts at 12:00:00 a.m. on the first day of the fiscal quarter after the current quarter ends, and continues for three months, ending at 11:59:59 on the last day of the fiscal quarter.
	XML example:
	`<query>{7.IR.'next fq'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'next+fq'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the fiscal quarter following the current fiscal quarter.

OPERATOR	DESCRIPTION

last fq	Starts at 12:00:00 a.m. on the first day of the fiscal quarter immediately preceding the current fiscal quarter and continues for three months, ending at 11:59:59 on the last day of the fiscal quarter.
	XML example:
	`<query>{7.IR.'last fq'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'last+fq'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the fiscal quarter immediately preceding the current fiscal quarter.

next N fq	Starts at 12:00:00 a.m. on first day of next fiscal quarter and continues for N fiscal quarters. (The range does not include the current fiscal quarter.)
	XML example:
	`<query>{7.IR.'next 5 fq'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'next+5+fq'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the 5 consecutive fiscal quarters starting the on the first day of next fiscal quarter.

OPERATOR	DESCRIPTION
last N fq	Starts at 12:00:00 a.m. N quarters before the current fiscal quarter and continues for N fiscal quarters. (The range does not include the current fiscal quarter.) XML example: `<query>{7.IR.'last 5 fq'}</query>` URL example: https://*target_domain*/db/*target_dbid*?a=API_DoQuery&query={7.IR.'last+5+fq'} This example returns: All records where the date or date/time in fid 7 falls within the 5 consecutive fiscal quarters that end immediately before the current fiscal quarter begins.

Relative date ranges: Years

A year is the 12-month time period beginning at 12:00:00 a.m. on January 1st and ending at 11:59:59 on December 31st. (Start and end times are always based on the application's time zone.)

OPERATOR	DESCRIPTION
this y	Starts at 12:00:00 a.m. January 1st of the current year and continues for 12 months (through December 31.) XML example: `<query>{7.IR.'this y'}</query>` URL example: https://*target_domain*/db/*target_dbid*?a=API_DoQuery&query={7.IR.'this+y'} This example returns: All records where the date or date/time in fid 7 occurs within the current year.

OPERATOR	DESCRIPTION

next y	Starts at 12:00:00 a.m. on the first day of the year immediately following the current year and continues for 12 months (through 11:59:59 p.m. on December 31).
	XML example:
	`<query>{7.IR.'next y'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'next+y'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the year following the current year.

last y	Starts at 12:00:00 a.m. on January 1st of the year immediately preceding the current year and continues for 12 months (through 11:59:59 p.m. on December 31.)
	XML example:
	`<query>{7.IR.'last y'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'last+y'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the year immediately preceding the current year.

OPERATOR	DESCRIPTION

next N y	Starts at 12:00:00 a.m. on January 1st of the year immediately following the current year and continues for N years. (The range does not include the current year.)
	XML example:
	`<query>{7.IR.'next 5 y'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'next+5+y'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the 5 consecutive years starting the on the first day of the year following the current year.

last N y	Starts at 12:00:00 a.m. N years before the current year and continues for N years. (The range does not include the current year.)
	XML example:
	`<query>{7.IR.'last 5 y'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'last+5+y'}`
	This example returns:
	All records where the date or date/time in fid 7 falls within the 5 consecutive years that end immediately before the current year begins.

Relative date ranges: Fiscal Years

Fiscal year definitions vary from organization to organization. For each Quick Base application, the application manager can define the fiscal year your company uses. You define the starting month of your fiscal year; the fiscal year continues for 12 months following the starting month. The fiscal year ends at 11:59:59 on the last day of the 12th month. (Start and end times are always based on the application's time zone.)

Quick Base automatically calculates your fiscal quarters based on the fiscal year starting month.

If your application manager does not set the fiscal year, Quick Base uses the calendar year as the default.

OPERATOR	DESCRIPTION

this fy	Starts at 12:00:00 a.m. on the first day of the current year and continues for 12 months.
	XML example:
	`<query>{7.IR.'this fy'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'this+fy'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the current fiscal year.

next fy	Starts at 12:00:00 a.m. on the first day of the fiscal year immediately following the current fiscal year and continues for 12 months.
	XML example:
	`<query>{7.IR.'next fy'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'next+fy'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the fiscal year following the current fiscal year.

last fy	Starts at 12:00:00 a.m. on the first day of the fiscal year that immediately precedes g the current fiscal year and continues for 12 months.
	XML example:
	`<query>{7.IR.'last fy'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'last+fy'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the fiscal year immediately preceding the current fiscal year.

OPERATOR	DESCRIPTION

next N fy	Starts at 12:00:00 a.m. on the start date of next fiscal year and continues for N years. (The range does not include the current fiscal year.)
	XML example:
	`<query>{7.IR.'next 5 fy'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'next+5+fy'}`
	This example returns:
	All records where the date or date/time in fid 7 occurs within the 5 consecutive fiscal years starting the on the first day of next fiscal year.

last N fy	Starts at 12:00:00 a.m. N fiscal years before the current fiscal year and continues for N fiscal years. (The range does not include the current fiscal year.)
	XML example:
	`<query>{7.IR.'last 5 fy'}</query>`
	URL example:
	`https://`*target_domain*`/db/`*target_dbid*`?a=API_DoQuery&query={7.IR.'last+5+fy'}`
	This example returns:
	All records where the date or date/time in fid 7 falls within the 5 consecutive fiscal years that end immediately before the current fiscal year begins.

Did this help you? Give us a rating:

API_DoQueryCount

Overview

You invoke this call against a table dbid using a query string to determine the number of records that will be returned.

Request parameters

PARAMETER	VALUE	REQUIRED?
query	The query string. See API_DoQuery for information on building the query string.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_DoQueryCount.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
numMatches	The number of records matching the query string.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_DoQueryCount

<qdbapi>
   <query>{'7'.XCT.'blue car'}</query>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
</qdbapi>
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_DoQueryCount</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <numMatches>1</numMatches>
</qdbapi>
```

Did this help you? Give us a rating:

API_EditRecord

Overview

Use API_EditRecord to change any editable field values in the specified record. Only those fields specified are changed; unspecified fields are left unchanged.

You can specify either field names or field IDs when using this call.

Request parameters

The following request parameters are available:

PARAMETER	VALUE	REQUIRED?
rid or key	The record ID of the record to be edited. You can obtain the recordID of any record in a query. (See API_DoQuery). If you use a key field OTHER than the record id (rid), you should use the key parameter rather than rid.	yes
update_id	Specifies the latest record state. Note that, if another user makes changes before you issue this call, the update ID in Quick Base will change, which will render invalid the update ID that you have. In this case, you should get the latest version of the record and its new update ID, and then make your changes. You can obtain the update ID for a record using the API_GetRecordInfo for the record you are editing.	no

PARAMETER	VALUE	REQUIRED?
field	Specify values for the fields that make up the record you want to add using either the **fid** attribute or **name** attribute of the <field> element. For example: `<field fid=”18”>Hi!</field> <field name=”Message”>Hi!</field>` You must specify all required fields. The values you enter here will vary, depending on field type. See How to specify values for different field types for more information.	yes
disprec	Set this parameter to 1 to specify that the new record should be displayed within the Quick Base application. An application login is required before the record can be displayed. If you use this parameter, Quick Base returns the normal Quick Base HTML page that displays the record. Omit this property if you don't want the new record to display within the Quick Base application.	no
fform	Set this parameter to 1 if you are invoking API_EditRecord from within an HTML form that has checkboxes and want those checkboxes to set Quick Base checkbox fields. Omit this property if you don't need Quick Base to set Checkbox fields based on your HTML page.	no
ignoreError	Set this parameter to 1 to specify that no error should be returned when a built-in field (for example, Record ID#) is written-to in an API_EditRecord call. If you do not set this parameter, Quick Base returns an error when API_EditRecord writes to a built-in field.	

PARAMETER	VALUE	REQUIRED?
ticket	Specifies valid authentication ticket. You obtain a valid ticket from API_Authenticate. A ticket is also returned in the various API responses. See API Overview for more information on the authentication ticket.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	Specifies a valid application token. See API Overview for more information on application tokens.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no
msInUTC	Allows you to specify that Quick Base should interpret all date/time stamps passed in as milliseconds using Coordinated Universal Time (UTC) rather than using the local application time. Set this parameter to 1 if you want to use Coordinated Universal Time. See usage example.	no
clist	A period-delimited list of field IDs to be returned. Quick Base will return values for these field IDs in the order in which you enter them here. To return all fields in a table, set this parameter to the value a. Allows you to return the state of dynamic fields like formulas, lookups or summaries without combining with API_DoQuery.	no

Response values

The response to this call contains the following:

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_AddField.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
rid	Record ID of the record that was edited
num_fields_changed	The number of record fields successfully updated in the edit.
update_id	This parameter is used to detect update conflicts when invoking API_EditRecord. You could save this update ID when you add a new record, but it would be better to instead get the most recent update_id value later when you query for the record to get it and update it.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_EditRecord
```

Using Field Names

```
<qdbapi>
  <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
     <rid>4</rid>
  <field name="start_date">06-08-2008</field>
   <field name="attendee_name">Joseph Riesentrotter</field>
</qdbapi>
```

Using Field IDs

```
<qdbapi>
<udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
     <rid>4</rid>
  <field fid="11">06-08-2008</field>
  <field fid="12">Joseph Riesentrotter</field>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_EditRecord&rid=154
&_fnm_second_year=1776&_fid_8=changed&update_id=992017018414
&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_EditRecord</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <ticket>2_bddqkpam2_dpsx_b_dhfjng8kqhysrb2vbisibpzjqz7</ticket>
   <rid>17</rid>
   <num_fields_changed>2</num_fields_changed>
   <update_id>1205700275470</update_id>
</qdbapi>
```

Did this help you? Give us a rating:

API_FieldAddChoices

Overview

Use this call to add new choices to a text - multiple choice or multi-select text field in a Quick Base table.

Note: Choices added to a Multi-select Text field are limited to 60 characters, and the total number of choices in the field may not exceed 100.

If you have Full Administration rights on the application, you can add new choices to any field of those types. If you don't have these rights, you can use this call only on fields with properties that are set to allow users to add new choices. (In the Quick Base UI, you can set the **Allow users to create new choices** field property to allow users to do this. You can also set this property using API_SetFieldProperties.)

If the choice you specify already exists in the list, it will not be added. No duplicates are allowed.

Request parameters

PARAMETER	VALUE	REQUIRED?
fid	The field ID of the setfield to which you want to add choices.	yes
choice	Specifies a choice you want to add. Use this tag multiple times to add more than one choice.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: - ticket - username/password - user token
apptoken	A valid application token.	yes, if the application requires application tokens.
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_FieldAddChoices.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
fid	The field ID of the multiple choice field to which you added choices.

ELEMENT NAME	VALUE
fname	The label of the field to which you added choices.
numadded	The number of choices successfully added.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_FieldAddChoices

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <fid>11</fid>
   <choice>Don Tomas</choice>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_FieldAddChoices&fid=7&choice=Bob
&choice=Mary&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_FieldAddChoices</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>mydata</udata>
   <fid>11</fid>
   <fname>Fumables</fname>
   <numadded>1</numadded>
</qdbapi>
```

Did this help you? Give us a rating:

API_FieldRemoveChoices

Overview

Use this call this to remove one or more of the options listed within a Text - Multiple Choice or Multi-select Text field in a Quick Base application. You can remove any choices that you created yourself. You must have Full Administration rights on the application to remove choices you did not create.

Request parameters

PARAMETER	VALUE	REQUIRED?
fid	The field ID of the field from which you want to remove choices.	yes
choice	Specifies a choice you want to remove. Use this tag multiple times to remove more than one choice. If any choice you specify cannot be removed (either because it does not exist, or because you lack permission to remove it), other valid choices you specify will be removed. The operation will be considered successful if at least one choice is removed. You should check the numremoved response value to make sure all the fields you expected were removed.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_FieldRemoveChoices.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
fid	The field ID of the multiple choice field from which you removed choices.

ELEMENT NAME	VALUE
fname	The label of the field from which you removed choices.
numremoved	The number of choices successfully removed. The call will be considered successful (errorcode=0) if one or more fields are removed successfully.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_FieldRemoveChoices

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
     <fid>11</fid>
   <choice>Black</choice>
  <choice>Blue</choice>
<qdbapi>
```

URL Alternative

```
https://target_domain/db/81768?a=API_FieldRemoveChoices&fid=11&choice=
black&choice=blue&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_FieldRemoveChoices</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>mydata</udata>
   <fid>11</fid>
   <fname>Color Choices</fname>
   <numremoved>2</numremoved>
</qdbapi>
```

Did this help you? Give us a rating:

API_FindDBByName

- Overview
- Request parameters
- Response values
- Sample XML Request
- URL alternative
- Sample response

Overview

Use API_FindDBByName to find the application-level dbid of an application whose name you know. When you use this call, Quick Base searches only those applications to which you have access. Because you can have multiple applications with the same name, you should be aware that more than one application dbid can be returned.

If a Quick Base application consists of a single table, this call returns the child dbid of that table (NOT the application dbid). If you want to find the application-level dbid in this case, set the **parentsOnly** parameter to 1 for the request.

Request parameters

PARAMETER	VALUE	REQUIRED?
dbname	The name of the application you want to find	yes
ParentsOnly	Ensures an app ID is returned, regardless of whether the application contains a single table or not. Note that, in applications that contain more than one table, Quick Base always returns the app dbid, regardless of whether you use this parameter or not. However, if an application contains only a single table, Quick Base returns the table DBID if you do not specify this parameter. If you are not sure whether your app contains more than one table, or if you are sure it contains only a single table, set this parameter to 1 to make sure an app dbid is returned.	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes

PARAMETER	VALUE	REQUIRED?
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_FindDBByName
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
dbid	Ensures an app ID is returned, regardless of whether the application contains a single table or not. Note that, in applications that contain more than one table, Quick Base always returns the app dbid, regardless of whether you using this parameter or not. However, if an application contains only a single table, Quick Base returns the table DBID if you do not specify this parameter. If you are not sure whether your app contains more than one table, or if you are sure it contains only a single table, set this parameter to 1 to make sure an app dbid is returned.
dbname	The application name, echoed in the response.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_FindDBByName

<qdbapi>
   <ticket>auth_ticket</ticket>
   <dbname>TestTable</dbname>
</qdbapi>
```

URL alternative

```
https://target_domain/db/main?a=API_FindDBByName&dbname=TestApp
&ticket=auth_ticket
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_FindDBByName</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <dbid>bdcagynhs</dbid>
</qdbapi>
```

Did this help you? Give us a rating:

API_GenAddRecordForm

- Overview
- Request parameters
- Response values
- Sample XML Request
- URL alternative

Overview

Use API_GenAddRecordForm to return the standard Quick Base **Add** *record* page for the table whose dbid you specify. The form contains edit fields for the user to complete and a **Save** button to add the record to the database.

If you want to pre-fill any fields, you can do so by supplying one or more field/value pairs in the request. Any fields not pre-filled or completed by the user are set to the default values set in the table field properties.

Request parameters

PARAMETER	VALUE	REQUIRED?
field	Specify the field name or fid to be pre-filled in the form and its value. Format the value as it is formatted in API_AddRecord. For example: • `<field name="Vehicle Make">Ford</field>`, or • `<_fid_11>1234</_fid_11>`	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
apptoken	A valid application token.	no
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

Returns an HTML page containing the **Add** *record* page with any prefilled values.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GenAddRecordForm

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <field name='email'>bada@bing.com</field>
</qdbapi>
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

URL alternative

```
https://target_domain/db/bdec4pdva?a=API_GenAddRecordForm&_fid_11=
GoodBye&_fid_23=Four%20score%20and%20seven%20years&ticket=auth_ticket
&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Did this help you? Give us a rating:

API_GenResultsTable

Overview

Use API_GenResultsTable to embed a Quick Base table report in an HTML page.

This call is typically used in its URL form embedded in an HTML page. When the HTML page is opened in a browser, the call runs automatically and returns results to qdbWrite() for display in the HTML page.

You can also specify that the results should be returned as a JavaScript array, in CSV format, or as tab separated values.

By default, API_GenResultsTable returns the same number of rows as would be returned in the UI. However, you can specify a different number of rows in your table by including **num-n** in the **options** parameter.

You can split a result set into several tables by calling API_GenResultsTable once for each table, using the **skp** option after the first invocation to skip over the records already returned.

Embedding a Quick Base report into a Web page

Use the following procedure to embed a Quick Base report into a Web page on your Web site:

1. In your Quick Base application, create the table report that you want to embed.
   Note: API_GenResultsTable should only be used with table reports.

2. In the HEAD section of your Web page, add the following <script> block:
   ```
   <script lang="javascript" src="https://target_domain/db/
   target_dbid?a=API_GenResultsTable&qid=5&jht=1&ticket=auth_ticket">
   </script>
   ```

   In this block, the *target_dbid* should be the table ID you want and *auth_ticket* should be replaced with a valid authentication ticket. (If you don't supply a ticket, the Quick Base login form appears when the page is opened and the user will have to log in first.) You can also supply your own query here, replacing the query id (qid) above with your own query id. See API_DoQuery for information on building a query.

3. In the HEAD section of your Web page, add the following style information:
   ```
   <style>
   td.m { font-family:verdana; font-size:70%; }
   td.hd { font-family:verdana; font-size:70%; font-weight:bold; color:white;}
   </style>
   ```

   By changing the attributes of these CSS tags, you can modify the format of the resulting table.

4. In the area of the Web page where you want the HTML table to display, add the following JavaScript code:

```
<script lang="javascript">
qdbWrite();
</script>
```

Request parameters

PARAMETER	VALUE	REQUIRED?
query or qid or qname	Specifies the query. You can use any of these three options: • Use `<query>{myfid.operator.value}</query>` to build your own query string. • Use `<qid>n</qid>`if you want to use a saved query and want to use its query ID. • Use `<qname>myQuery</qname>` if you want to use a saved query and refer to it by its query name. • See API_DoQuery for information on building your own query.	no
clist	A period-delimited list of field IDs to be returned. Quick Base will return values for these field IDs in the order in which you enter them here. To return all fields in a table, set this parameter to the value a. Omit this parameter if you want the query to return the table's default columns.	no

PARAMETER	VALUE	REQUIRED?

slist	A period-delimited list of field IDs used to determine sorting as follows:
	`<slist>fid</slist>`
	The following slist parameter sorts all records by the field whose fid is 7.
	`<slist>7</slist>`
	You can specify a secondary sort by including another fid in the string (separating each fid with a period)
	`<slist>7.12</slist>`
	URL example:
	`&slist=7.12`
	If you don’t specify the sort list, the default sort specified on the Report Layout page determines the order in which records are displayed.
		no

jht	Returns the HTML for a table as a JavaScript function named `qdbWrite()`. You can then embed the HTML table into a Web page by calling the JavaScript function in the area of the Web page where you want the table to display. Set the value of the parameter to ‘n’ if you want the new look and feel introduced on Apr-12-2003. Set the value of this parameter ‘1’ if you want the CSS styles that render the HTML table with the Quick Base look and feel prior to Apr-11-2003.	yes
jsa	Set this parameter to 1 if you want the results returned as a JavaScript array.	no

PARAMETER	VALUE	REQUIRED?
options	A period-delimited list containing any or any combination of the following options: • **num-n** return a maximum of n records Note that num-n cannot be combined with the csv or tsv options. • **onlynew** return only those records marked with new or updated flags • **skp-n** skip the first n records returned by the query Note that skp-n cannot be combined with the csv or tsv options. • **sortorder-A** for ascending sorts • **sortorder-D** for descending sorts • **ned** to omit the edit icons in HTML table format • **nvw** to omit the view icons in HTML table format • **nfg** to omit the new and updated icons in HTML table format • **phd** for plain (non-hyperlinked) headers • **abs** for absolute URLs • **csv** for comma-separated value output format Note that csv cannot be combined with the num-n or skp-n options. • **tsv** for tab-separated value output format Note that tsv cannot be combined with the num-n or skp-n options.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

If you specify the **jht** option, the results are written to `qbdWrite()` embedded in the HTML page. Optionally, if you specify **CSV** or **tab** in the options parameter, the return will be comma separated values (CSV) or tab separated values, respectively. If you specified **jsa**, the JavaScript array is returned.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GenResultsTable

qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <query>{'11'.CT.'Bob'}AND{'19'.GTE.'5'}</query>
   <clist>6.7.9.11.16</clist>
   <slist>11.6</slist>
   <fmt>structured</fmt>
   <options>num-4.sortorder-D</options>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_GenResultsTable&ticket=
auth_ticket&apptoken=app_token&qid=5&jht=1&query=
{'6'.CT.'done'}AND{'14'.CT.'USA'}&clist=7.8&slist=6&options=sortorder-A
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

A Sample Web Page HTML with an Embedded Table

```
<html>
<head>
<script lang="javascript"
src="https://quickbase.com/db/5rh82jcq?a=API_GenResultsTable
      &ticket=auth_ticket&apptoken=app_token
      &qid=5&jht=1">
</script>
<style>
   td.m { font-family:verdana; font-size:70%; }
   td.hd { font-family:verdana; font-size:70%; font-weight:bold;
    color:white;}
</style>
</head>
<body>
<h1>Example</h1>
   <table cellpadding=5 bgcolor=lightgreen>
      <tr>
         <td valign=top nowrap><b>A Quick Base table<br>embedded in
            a<br>page of HTML
         </td>
         <td>
            <script lang="javascript">
               qdbWrite();
            </script>
         </td>
      </tr>
   </table>
</body>
</head>
</html>
```

Did this help you? Give us a rating:

API_GetAncestorInfo

Overview

This call must be invoked on the app dbid (not on a table dbid) to get ancestor information about the current application. To use this call, you must have at least read permissions on the table.

The first generation copy of an application returns the same DBID for both oldestancestorappid and ancestorappid. The grandchild of a template (and subsequent generations) shows the immediate parent app as the ancestorappid and the original template as the oldestancestorappid.

Request parameters

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	Supply a valid application token.	yes, if the application requires application tokens

Response values

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_AddField.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
ancestorappid	dbid for the ancestor app.
oldestancestorappid	dbid for the master application from which the current application was ultimately copied. For immediate copies, the ancestorappid and the oldestancestorappid are the same.

Sample XML Request

```
POST https://target_domain/db/target_dbid
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION:API_GetAncestorInfo

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
</qdbapi>
```

URL Alternative

```
https://target_domain/db/target_dbid?a=API_GetAncestorInfo
&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample Response

```
<?xml version="1.0" ?>
  <qdbapi>
    <action>api_getancestorinfo</action>
    <errcode>0</errcode>
    <errtext>No error</errtext>
    <ancestorappid>bbyhxrmsv</ancestorappid>
    <oldestancestorappid>bbyhxrmsv</oldestancestorappid>
  </qdbapi>
```

Did this help you? Give us a rating:

API_GetAppDTMInfo

- Overview
- Request parameters
- Response values
- URL alternative
- Sample Response using the app dbid
- Sample Response using the table dbid

Overview

Use API_GetAppDTMInfo to get the timestamp of the last time change throughout the application. You must invoke this call on an application dbid, not on a table dbid.

This is a fast, unobtrusive way to get the time of the last change in an application schema or in records within its tables. No ticket or app token is required for this call, nor does this call result in the application being loaded into memory in Quick Base.

The response to this call contains a parameter indicating the next time you will be allowed to make this call again.

Request parameters

PARAMETER	VALUE	REQUIRED?
dbid	The application dbid.	yes

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_GetAppDTMInfo.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.

ELEMENT NAME	VALUE
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
RequestTime	The time at which the server received this request. All times expressed as milliseconds since January 1st 1970 00:00:00, UTC.
RequestNextAllowedTime	The earliest time that another API_GetDBDTMInfo request for the same Application DB ID is permitted. All times expressed as milliseconds since January 1st 1970 00:00:00, UTC. If another request is received before this time, an errcode of 77 is returned, with an errtext of "API request limit exceeded".

app	Aggregate containing the last time the application schema was modified (lastModifiedTime) and the last time the application’s main table was written to (lastRecModTime).
	Invoke this call on an application dbid. The aggregate is structured as follows:
	`<app id="the app dbid here">`
	`    <lastModifiedTime>milliseconds</lastModifiedTime>`
	`    <lastRecModTime>milliseconds</lastRecModTime>`
	` </app>`
	All times expressed as milliseconds since January 1st 1970 00:00:00, UTC.

tables	Aggregate containing one or more table sub-aggregates. The table sub-aggregate contains the following:
	`<table id="the table dbid here">`
	`    <lastModifiedTime>milliseconds</lastModifiedTime>`
	`    <lastRecModTime>milliseconds</lastRecModTime>`
	` </table>`
	where lastModifiedTime refers to any changes made to the table schema and lastRecModTime refers to any writes to the table in terms of records added or modified.
	All times expressed as milliseconds since January 1st 1970 00:00:00, UTC.

URL alternative

`https://`*target_domain*`/db/main?a=API_GetAppDTMInfo&dbid=`*target_dbid*

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response (using app dbid)

```
<?xml version="1.0" ?>
<qdbapi>
<action>API_GetAppDTMInfo</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<RequestTime>1227660044062</RequestTime>
<RequestNextAllowedTime>1227660049062</RequestNextAllowedTime>
<app id="bdzk2ecg5">
<lastModifiedTime>1227657049750</lastModifiedTime>
<lastRecModTime>1227647748330</lastRecModTime>
</app>
<tables>
<table id="bdzk2ecg6">
<lastModifiedTime>1227647748440</lastModifiedTime>
<lastRecModTime>1227647748330</lastRecModTime>
</table>
<table id="bdzma8u8r">
<lastModifiedTime>1227657006030</lastModifiedTime>
<lastRecModTime>1227657006030</lastRecModTime>
</table>
<table id="bdzma97vj">
<lastModifiedTime>1227657049720</lastModifiedTime>
<lastRecModTime>1227657049720</lastRecModTime>
</table>
</tables>
</qdbapi>
```

Sample response (using table dbid)

```
<?xml version="1.0" ?>
<qdbapi>
<action>API_GetAppDTMInfo</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<RequestTime>1227726628024</RequestTime>
<RequestNextAllowedTime>1227726633024</RequestNextAllowedTime>
<tables>
<table id="bdzpdhxua">
<lastModifiedTime>1227726509647</lastModifiedTime>
<lastRecModTime>1227726509647</lastRecModTime>
</table>
</tables>
</qdbapi>
```

Did this help you? Give us a rating:

API_GetDBPage

- Overview
- Request parameters
- Response values
- Sample XML Request
- URL alternative

Overview

Use API_GetDBPage to retrieve a stored page in Quick Base. Quick Base allows you to store various types of pages, ranging from user-guide pages for your application to Exact Forms, used to automate insertion of data into Word documents using a special Word template from Quick Base. (See the Quick Base online help for more information about DB pages.)

Request parameters

PARAMETER	VALUE	REQUIRED?
pageID	The ID of the page. You can also use the pagename.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens

PARAMETER	VALUE	REQUIRED?
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

The requested page is returned in HTML.

XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GetDBPage

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <pageID>3</pageID>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_GetDBPage&pageid=18&ticket=auth_ticket&apptok
en=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Did this help you? Give us a rating:

API_GetDBInfo

Overview

Use API_GetDBInfo to get metadata information, such as the last time the table was modified. For example, you might use this function to find out if the table has changed since you last used it, or to find out if a new record has been added to the table. You can invoke this call on either an application-level dbid or a table-level dbid.

Request parameters

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	A user token – this is a ticket alternative, used for API access	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_GetDBInfo.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
dbname	The name of the application or table.
lastRecModTime	The last time a user modified a record.
lastModifiedTime	The last time a user modified the table.
createdTime	The time the table was created.
numRecords	The number of records in the table.
mgrID	A number that uniquely identifies the table manager.
mgrName	The name of the table manager.
version	Reserved.

ELEMENT NAME	VALUE
time_zone	A string value that defines the application's time zone.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GetDBInfo

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_GetDBInfo&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_GetDBInfo</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <dbname>test</dbname>
   <lastRecModTime>1205806751959</lastRecModTime>
   <lastModifiedTime>1205877093679</lastModifiedTime>
   <createdTime>1204745351407</createdTime>
   <numRecords>3</numRecords>
   <mgrID>112149.bhsv</mgrID>
   <mgrName>AppBoss</mgrName>
   <version>2.0</version>
   <time_zone>(UTC-08:00) Pacific Time (US & Canada)</time_zone>
</qdbapi>
```

Did this help you? Give us a rating:

API_GetDBvar

- Overview
- Request parameters
- Response values
- Sample XML Request
- URL alternative

Overview

Use API_GetDBVar to get values assigned to variables (DBVars) set up in your Quick Base application. DBVars are variables you can create to use in your application. (See the Quick Base online help to learn how to set up DBVars.

API_GetDBVar returns only one DBVar. If you want to retrieve more than one variable, you'll need to invoke this call multiple times.

Once you've set them up, you can assign values to them using API_SetDBVar. (Use the application-level dbid.) You'll get an error if the DBVar you ask for does not exist.

Request parameters

PARAMETER	VALUE	REQUIRED?
varname	The name of the variable in the target application.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	Supply a valid application token.	yes, if the application requires application tokens

PARAMETER	VALUE	REQUIRED?
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_AddField.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
value	The value of the DBVar.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GetDBVar

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <varname>usercode</varname>
</qdbapi>
```

URL alternative

`https://`*target_domain*`/db/`*target_dbid*`?a=API_GetDBVar&ticket=`*auth_ticket*
`&apptoken=`*app_token*`&varname=usercode`

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*.
Read about this notation.

Sample XML Response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_getDBvar</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <value>12</value>
</qdbapi>
```

Did this help you? Give us a rating:

API_GetFieldProperties

Overview

Use API_GetFieldProperties to view the properties of a field. Returns properties specific to the field type.

API_GetFieldProperties lists summary field metadata. For example, Total Actual Hours is a summary field defined in a project table representing the actual number of hours team members have logged against tasks on time cards. It is based on the following field definitions:

- It uses a relationship based on a reference field “related project” (fid: 48), and the API returns it as summaryReferenceFid

- The field that it is summarizing is Actual Hours (fid: 133), and the API returns it as summaryTargetFid

- The summary function is Total, and the API returns it as is

The sample code displays as follows:

```
<field>
   <summaryReferenceFid>48</summaryReferenceFid>
   <summaryTargetFid>133</summaryTargetFid>
   <summaryFunction>Total</summaryFunction>
 </field>
```

Request parameters

PARAMETER	VALUE	REQUIRED?
fid	field id	yes

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_GetFieldProperties.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1Content-Type: application/xmlContent-
Length: QUICKBASE-ACTION:API_GetFieldProperties

<qdbapi><fid>35</fid></qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/target_dbid?a=API_GetFieldProperties&fid=35
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

Returns field properties based on the field type. The following example shows the response for a Formula - URL field type.

```
<?xml version="1.0"
?><qdbapi>    <action>API_GetFieldProperties</action>    <errcode>0</errcode>
<errtext>No error</errtext>       <field id=”35” field_type=”url”
base_type=”text” mode=”virtual”>      <label>Add URL
Action</label>       <nowrap>0</nowrap>       <bold>0</bold>       <required>0</
required>        <appears_by_default>1</appears_by_default>       <find_enabled
>0</find_enabled>        <allow_new_choices>0</allow_new_choices>       <sort_as
_given>1</sort_as_given>       <formula> URLRoot () &”db/” & [_DBID_URL_ACTIONS]
& “?a=API_GenAddRecordForm&_fid_7=” & URLEncode ([Record ID#]}& “&z=” &
Rurl()       <BR/>       </formula>      <carrychoices>1</carrychoices>      <fo
reignkey>0</</foreignkey>      <unique>0</unique>      <doesdatacopy>0</doesdata
copy>      <fieldhelp/>       <appears_as>Add URL
```

```
action</appears_as>      <abbreviate>0</abbreviate>      <use_new_window>0</use_
new_window>      <auto_save>0</auto_save>     </field></qdbapi>
```

Did this help you? Give us a rating:

API_GetGroupRole

Overview

Use API_GetGroupRole determine the roleid for a group within an application.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the group to manage.	yes
apptoken	A valid application token.	yes, if the application requires application tokens.
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_GetGroupRole.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
roles	The roles the group has in the app.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GetGroupRole

<qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <gid>345889.klsd</gid>
</qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/target_dbid?a=API_GetGroupRole&gid=345889.klsd&
ticket=auth_ticket&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?><qdbapi>
   <action>API_GetGroupRole</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
   <roles>
      <role id=”11”><name>Participant</name>
   </role>
</roles>
</qdbapi>
```

Did this help you? Give us a rating:

API_GetNumRecords

Overview

Use API_GetNumRecords to get the total number of records in the table. Invoke this call on a table-level dbid.

Request parameters

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if application tokens are being used by the application
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_AddField.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
num_records	The total number of records in the table.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GetNumRecords

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
</qdbapi>
```

URL Alternative

```
https://target_domain/db/target_dbid?a=API_GetNumRecords
&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_GetNumRecords</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <num_records>17</num_records>
</qdbapi>
```

Did this help you? Give us a rating:

API_GetRecordAsHTML

Overview

Use API_GetRecordAsHTML to get a record within an HTML fragment that can be embedded in another Web page.

Request parameters

PARAMETER	VALUE	REQUIRED?
rid	The record ID of the record to be edited. You can obtain the recordID of any record in a query. (See API_DoQuery.) If you use a key field OTHER than the record id (rid), you should use the key parameter rather than rid.	yes
dfid	The dform id of the form used to generate the HTML. The resulting display matches the contents and layout of the form you specify. You can obtain the dfid of a form by previewing or editing the form; the dfid is part of the URL in the browser's address bar.	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: - ticket - username/password - user token
apptoken	Supply a valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

The requested record, in an HTML fragment.

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_AddField.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GetRecordAsHTML

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <rid>2</rid>
   <dfid>10</dfid>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_GetRecordAsHTML&rid=18&dfid=10
&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Embedding a Quick Base record into a Web page

To embed a Quick Base record into a Web page on your site, add the following code to the body of your Web page:

```
<div id="dbPagePayload"></div>
<script lang="text/javascript">
$.post("./target_dbid?a=API_GetRecordAsHTML&rid=n", {}, function(response){
$("#dbPagePayload").html(response);
});
</script>
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Example Web Page Code

```
<html>
<head>
<!-- put styling information here -->
</head>
<body>
<h1>Example</h1>
<p> A Quick Base table embedded in a page of HTML</p>
<div id="dbPagePayload"></div>
<script lang="text/javascript">
$.post("./target_dbid?a=API_GetRecordAsHTML&rid=20", {}, function(response){
$("#dbPagePayload").html(response);
});
</script>
</body>
</head>
</html>
```

Did this help you? Give us a rating:

API_GetRecordInfo

Overview

Use API_GetRecordInfo to get the values and types of all the fields in a record. You invoke this call on a table-level dbid.

Request parameters

PARAMETER	VALUE	REQUIRED?
rid or key	The record ID of the record to be edited. You can obtain the record ID of any record in a query. (See API_DoQuery.) If you use a key field OTHER than the record id (rid), you should use that field rather than rid when you call API_GetRecordInfo.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via an API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens

PARAMETER	VALUE	REQUIRED?
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_GetRecordInfo.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
rid	The record ID.
num_fields	The total number of fields in this record, including built-in fields.
update_id	The update ID, to be used in editing the field.

ELEMENT NAME	VALUE

<field> aggregates	Several (num_field many) field aggregates will be returned. All record fields will be returned, even the standard built-in fields. The aggregate is structured as follows:
	`<field>`
	`    <fid>9</fid>`
	`    <name>URL</name>`
	`    <type>URL</type>`
	`    <value>www.example.com</value>`
	` </field>`
	Some fields may have additional elements.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GetRecordInfo

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <rid>2</rid>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_GetRecordInfo&rid=18
&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
<action>api_getrecordinfo</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<udata>mydatamin</udata>
<rid>20</rid>
<num_fields>28</num_fields>
<update_id>1205780029699</update_id>
<field>
<fid>6</fid>
```

```
<name>Start Date</name>
<type>Date</type>
<value>1437609600000</value>
<printable>23-07-15</printable>
</field><field>
<fid>7</fid>
<name>File</name>
<type>File Attachment</type>
<value>BatchID.html</value>
</field>
.
.
.
</qbdapi>
```

Did this help you? Give us a rating:

API_GetRoleInfo

Overview

Use API_GetRoleInfo to get all of the roles that apply to the application. You invoke this call on the application-level dbid.

The access level returned in the <access> field is one of these available access types:

- Basic Access (able to view/add/modify records, depending on permissions)
- Basic Access with Share (same as Basic, but can share the application with other users)
- Administrator (full administrative access)

Request parameters

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token	yes, if the application requires application tokens

PARAMETER	VALUE	REQUIRED?
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_GetRoleInfo.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
roles	An aggregate containing one or more roles, structured as follows: `<roles>  <role id="11">       <name>Participant</name>        <access id="3">Basic Access</access>    </role>  </roles>` Each role returned has a role ID as an attribute and contains a role name and the application access level that is conferred by the role.
udata	Optional. A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
```

```
Content-Length:
QUICKBASE-ACTION: API_GetRoleInfo

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_GetRoleInfo&apptoken=auth_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_GetRoleInfo</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <roles>
      <role id="10">
         <name>Viewer</name>
         <access id="3">Basic Access</access>
      </role>
      <role id="11">
         <name>Participant</name>
         <access id="3">Basic Access</access>
      </role>
      <role id="12">
         <name>Administrator</name>
         <access id="1">Administrator</access>
      </role>
   </roles>
</qdbapi>
```

Did this help you? Give us a rating:

API_GetSchema

Overview

Use API_GetSchema to get information about the specified application or application table.

When invoked on an application dbid, this call returns DBVars created for the application and all child table dbids.

When invoked on a table dbid, the call returns DBVars and additional table-related information such as reports, field IDs (fid), key field ID (key_fid), individual record names, base type, and the current property settings for each field. In addition, this call returns the default List All and List Changes reports.

This call also returns any custom reports that you have created in the UI and saved as a common report. You can then use the query ID or query name in the various API calls that use queries, instead of manually constructing the query string yourself. See API_SetFieldProperties and API_DoQuery for more information on field property values.

Here's a sample of what one of these saved queries looks like in the response:

```
<query id="5">
   <qname>Complete Refuel</qname>
   <qytype>table</qytype>
   <qydesc>fuel quantity greater than 10 gallons</qydesc>
   <qycrit>({'7'.AF.'03-24-2008'}AND{'6'.EX.'10'})</qycrit>
   <qyclst>8.6</qyclst>
   <qyopts>nos.</qyopts>
   <qycalst>0.0</qycalst>
</query>
```

This call can be used to create tooltips for your application based on the field help in your Quick Base application. You can use API_GetSchema to get the field help and then load it into tooltips.

API_GetSchema lists summary field metadata. For example, Total Actual Hours is a summary field defined in a project table representing the actual number of hours team members have logged against tasks on time cards. It is based on the following field definitions:

- It uses a relationship based on a reference field “related project” (fid: 48), and the API returns it as summaryReferenceFid

- The field that it is summarizing is Actual Hours (fid: 133), and the API returns it as summaryTargetFid

- The summary function is Total, and the API returns it as is

The sample code displays as follows:

```
<field>
   <summaryReferenceFid>48</summaryReferenceFid>
   <summaryTargetFid>133</summaryTargetFid>
   <summaryFunction>Total</summaryFunction>
 </field>
```

Request parameters

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_GetSchema.

ELEMENT NAME	VALUE
date_format	The date configuration for the app or the table. Date formats supported are: • MM-DD-YYYY • MM-DD-YY • DD-MM-YYYY • DD-MM-YY • YYYY-MM-DD Note that each element of the date format is separated by a dash (-), and dates are saved in this format.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
time_zone	A string value that defines the application's time zone.
udata	Optional. Contains any udata value supplied in the request.
table	Contains all of the application or table metadata, depending on the dbid supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GetSchema

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_GetSchema&ticket=auth_ticket
&apptoken=app_token&udata=mydata
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample Application-level Response

```
<?xml version="1.0" ?>
<qdbapi>
     <action>API_GetSchema</action>
     <errcode>0</errcode>
     <errtext>No error</errtext>
     <time_zone>(UTC-08:00) Pacific Time (US & Canada)</time_zone>
      <date_format>YYYY-MM-DD</date_format>
      <udata>mydata</udata>
      <table>
          <name>API created Sample</name>
          <desc>This is a sample application.</desc>
          <original>
                <app_id>bdb5rjd6h</app_id>
                <table_id>bdb5rjd6h</table_id>
                <cre_date>1204586581894</cre_date>
                <mod_date>1206394201119</mod_date>
                <next_record_id>1</next_record_id>
                <next_field_id>7</next_field_id>
                <next_query_id>5</next_query_id>
                <def_sort_fid>6</def_sort_fid>
                <def_sort_order>1</def_sort_order>
          </original>
          <variables>
               <var name="Blue">14</var>
               <var name="Jack">14</var>
               <var name="Magenta">12</var>
               <var name="usercode">14</var>
          </variables>
          <chdbids>
              <chdbid name="_dbid_doug_s_api_created_sample">bdb5rjd6g</chdbid>
              <chdbid name="_dbid_vehicle">bddrydqhg</chdbid>
          </chdbids>
         <fields></fields>
      </table>
</qdbapi>
```

Sample Table-level Response

```
<?xml version="1.0" ?>
<qdbapi>
 <action>API_GetSchema</action>
 <errcode>0</errcode>
 <errtext>No error</errtext>
 <time_zone>(UTC-08:00) Pacific Time (US & Canada)</time_zone>
 <date_format>YYYY-MM-DD</date_format>
 <udata>mydata</udata>
 <table>
   <name>API created Sample</name>
   <desc>This is a sample contact manager table.</desc>
   <original>
     <table_id>bdb5rjd6g</table_id>
     <app_id>bdb5rjd6g</app_id>
     <cre_date>1204586581894</cre_date>
     <mod_date>1206394201119</mod_date>
     <next_record_id>22</next_record_id>
     <next_field_id>24</next_field_id>
     <next_query_id>5</next_query_id>
     <def_sort_fid>6</def_sort_fid>
     <def_sort_order>1</def_sort_order>
   </original>
   <variables>
     <var name="Blue">14</var>
     <var name="Jack">14</var>
     <var name="Magenta">12</var>
     <var name="usercode">14</var>
   </variables>
   <queries>
    <query id="1">
     <qyname>List All</qyname>
      <qytype>table</qytype>
      <qycalst>0.0</qycalst>
     </query>
     <query id="2">
      <qyname>List Changes</qyname>
      <qytype>table</qytype>
      <qydesc>Sorted by Date Modified</qydesc>
      <qyslst>2</qyslst>
      <qyopts>so-D.onlynew.</qyopts>
      <qycalst>0.0</qycalst>
     </query>
   </queries>
   <fields>
     <field id="6" field_type="text" base_type="text">
     <label>Additional Information</label>
     <fieldhelp>Supply additional data about the problem.</fieldhelp>
      <nowrap>0</nowrap>
      .
      .
      .
    </field>
   </fields>
 </table>
</qdbapi>
```

Did this help you? Give us a rating:

API_GetUserInfo

Overview

Use API_GetUserInfo to get the username and user ID associated with an email address used for Quick Base sign in. You invoke this call on db/main. You can use this call to grant a user access rights to your application and then invite that user to your application. This call is typically made to return the Quick Base user ID for a user whose email address you know, in preparation for subsequent calls to API_AddUserToRole (grant access rights) and API_SendInvitation, both of which require the user ID.

The user email that you specify must be recognized in Quick Base. (For users who aren’t registered in Quick Base yet, use the alternative call API_ProvisionUser.)

For anonymous users, this call returns:

```
<user id="1.ckbs">
<firstName></firstName>
<lastName></lastName>
<login>anonymous</login>
<email></email>
<screenName>anonymous</screenName>
<isVerified>0</isVerified>
<externalAuth>0</externalAuth>
</user>
```

If the anonymous user passes in an **email** parameter, the email will be passed back in the response.

If you don’t supply an **email** parameter, the **ticket** parameter will be used to determine the current user, which means the current user ID, name, and email are returned.

Request parameters

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token	yes, if the application requires application tokens
email	Supply the email address (as registered with Quick Base) of the user whose information you want. You can also supply the user's user name. If you don't supply this parameter, the call returns the user info of the user to whom the ticket was issued (you). For anonymous users, the email parameter is ignored.	no
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_AddField.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.

ELEMENT NAME	VALUE
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
ticket	The ticket needed for application access.

user	The returned user aggregate. This aggregate always contains the user ID attribute and the user name. The user aggregate also contains the user login if the **email** parameter was NOT specified in the request. The login will be any of the following:
	• The user's LDAP login, if applicable
	• The user's Quick Base screen (user) name, if LDAP is not being used
	• The user's email address, if the user has no LDAP login or Quick Base user name
	The aggregate is structured as follows:
	`<user id="56760756.bpxd">`
	`     <firstName>jdoe</firstName>`
	`     <lastName>qb</lastName>`
	`     <login>jdoe.qb@gmail.com</login>`
	`     <email>jdoe.qb@gmail.com</email>`
	`     <screenName></screenName>`
	`      <isVerified>1</isVerified>`
	`       <externalAuth>0</externalAuth>`
	` </user>`

udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GetUserInfo

<qdbapi>
   <ticket>auth_ticket</ticket>
   <email>Ragnar-Lodbrok@paris.net</email>
   <udata>optional data</udata>
</qdbapi>
```

URL alternative

```
https://target_domain/db/main?a=API_GetUserInfo&ticket=auth_ticket
&email=Ragnar-Lodbrok@paris.net
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>api_getuserinfo</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <user id="112149.bhsv">
      <firstName>Ragnar</firstName>
      <lastName>Lodbrok</lastName>
      <login>Ragnar</login>
      <email>Ragnar-Lodbrok@paris.net</email>
      <screenName>Ragnar</screenName>
   </user>
</qdbapi>
```

Did this help you? Give us a rating:

API_GetUserRole

Overview

Use API_GetUserRole to see what roles are assigned to a specific user in an application. Invoke this call on an application-level dbid.

Note that, if you do not have sharing permissions on the application, the API call returns your own roles only. To get your roles, you must either invoke the call with no user ID or supply your own user ID.

You must have Basic Access with Sharing or Full Administration permissions to use this API call to get roles for users other than yourself. Also, note that, if you want this call to return any roles assigned to a user because he or she is part of a group, you must use the **inclgrps** parameter (described below). If you do not specify this parameter, API_GetUserRole returns only those roles assigned to the individual user.

Request parameters

PARAMETER	VALUE	REQUIRED?
userid	The user ID of the user whose current role you want to retrieve. You can get the userid using API_GetUserInfo. If you want to get your own roles, you can use your own userid or omit this parameter. To get the roles of another user, you must supply the user's user id. You must also have Basic Access with Sharing or Full Administration permissions on the application.	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
inclgrps	Set this parameter to 1 if you want the call to return roles assigned to groups to which the user belongs. The call will return additional information about the role's type--that is whether the role is a user role, a role assigned to a group role, or a role assigned to a domain group. (See *Response Values*, below.)	no
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_GetUserRole.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.

<table>
<tr><th>ELEMENT NAME</th><th>VALUE</th></tr>
<tr><td>errtext</td><td>Text that explains the error code.
"No error" indicates that no error was encountered.</td></tr>
<tr><td>user</td><td>The userid is returned as an attribute in this aggregate. The user aggregate also contains these parameters:
<ul>
<li>name -- the user name</li>
<li>roles -- an aggregate listing all roles to which the user belongs. The roles aggregate specifies these parameters:
<ul>
<li>role id -- the ID of the role</li>
<li>name -- the name of the role</li>
<li>access id -- the application access level for the role</li>
<li>member -- appears only if you've used inclgrps in the request. This parameter specifies the type of role. You can specify an attribute of “type.” Possible values are
<ul>
<li>user -- role is assigned to a user</li>
<li>group -- role is assigned to a group</li>
<li>domainGroup -- role is assigned to a domain group</li>
</ul></li>
</ul></li>
</ul>
The following is an example of the user response:
<pre>
<user id="112245.efy7">
    <name>John Doe</name>
    <roles>
      <role id="11">
      <name>Participant</name>
      <access id="3">Basic Access</access>
      <member type="user">John Doe</member>>
      </role>
    </roles>
 </user>
</pre></td></tr>
<tr><td>udata</td><td>Optional. Contains any udata value supplied in the request.</td></tr>
</table>

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length: 77
QUICKBASE-ACTION: API_GetUserRole

<qdbapi>
```

```
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <inclgrps>1</inclgrps>
  <userid>112245.efy7</userid>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_GetUserRole&ticket=auth_ticket
&apptoken=app_token3&inclgrps=1&userid=112245.efy7
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_GetUserRole</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <user id="112245.efy7">
   <name>John Doe</name>
      <roles>
         <role id="11">
           <name>Participant</name>
           <access id="3">Basic Access</access>
           <member type="user">John Doe</member>
         </role>
         <role id="10">
           <name>Viewer</name>
           <access id="3">Basic Access</access>
           <member type="group">Group1</member>
         </role>
         <role id="12">
           <name>Administrator</name>
           <access id="1">Administrator</access>
           <member type="domainGroup">example.com</member>
         </role>
      </roles>
   </user>
</qdbapi>
```

Did this help you? Give us a rating:

API_GetUsersInGroup

Overview

Use API_GetUsersInGroup to get the list of users and groups that make up the members and/or managers of a group.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the group to list.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes
includeAllMgrs	True or false (default). When this is set to **true**, both members and managers of the group are returned.	no
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_GetUsersInGroup.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
users	A list of the users in the group.
managers	A list of the managers in the group.
subgroups	A list of the subgroups in the group.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION:API_GetUsersInGroup

<qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <includeAllMgrs>true</includeAllMgrs>
   <gid>2345.skdj</gid>
</qdbapi>
```

Sample HTTP Get Request

https://*target_domain*/db/main?a=API_GetUsersInGroup&gid=2345.skdj&includeAllMgrs=true&ticket=*auth_ticket*&udata=misc%20data

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_GetUsersInGroup</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
   <group id=”2345.sdfk”>
    <name>GroupInfoTestGroup</name>
    <description>My Group description</description>
    <users>
    <user id="112149.bhsv">
       <firstName>john</firstName>
       <lastName>doe</lastName>
       <email>jdoe.qb@gmail.com</email>
       <screenName></screenName>
       <isAdmin>false</isAdmin>
    </user>
    <user id="192156.mqbr">
       <firstName>justin</firstName>
       <lastName>sample</lastName>
       <email>justin@example.com</email>
       <screenName></screenName>
       <isAdmin>false</isAdmin>
    </user>
    </users>
    <managers>
    <manager id="52731770.b82h">
       <firstName>Angela</firstName>
       <lastName>Leon</lastName>
       <email>angela_leon@aleon.com</email>
       <screenName>aqleon</screenName>
       <isMember>true</isMember>
    </manager>
    </managers>
    <subgroups>
    <subgroup id=”3450.aefs”/>
    </subgroups>
   </group>
</qdbapi>
```

Did this help you? Give us a rating:

API_GrantedDBs

- Overview
- Request parameters
- Response values
- Sample XML Request
- URL alternative
- Sample response

Overview

Use API_GrantedDBs to get a list of the names and dbids of all the applications and tables that you are entitled to access, across all domains. You invoke this call on https://*target_domain*/db/main.

This call returns all the apps that you have access to, across all domains, not only the target domain. However, you can use the *realmAppsOnly* parameter to make this call return only the apps in the realm the call is being made against.

If you specify that both parent and child tables are to be returned, you can easily identify the parent from child by the database name that is shown. Table names are appended at the end of application names, as shown below:

Application name: MyApp
Child table name: MyApp:MyTable

Request parameters

PARAMETER	VALUE	REQUIRED?
adminOnly	Optional. Set this paramter to "true" to return only tables where the user making the request has administration privileges.	no
excludeparents	Specifies whether you want application-level dbids returned. Valid values are: 0—Return application-level dbids. (Default) 1—Do not return application-level dbids.	no
includeancestors	Set this parameter to 1 to include ancestor and oldest ancestor information. Omit this property if you don't want to see ancestor information.	no

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no
withembeddedtables	Specifies whether you want child table dbids to be returned Valid values are: 0—Do not return child table dbids 1—Return child table dbids (Default)	no
realmAppsOnly	Set this parameter to "true" to return only the apps in the realm the call is being made against.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_AddField.

ELEMENT NAME	VALUE
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.

databases	This aggregate contains the dbinfo sub-aggregate. There could be 0 (zero) or more of these returned in the response. A dbinfo sub-aggregate is structured as follows:
	`<dbinfo>`
	`    <dbname>MyDatabase</dbname>`
	`    <dbid>bcaour4ak</dbid>`
	` </dbinfo>`
	The returned dbinfos could be applications only (no child tables) or child tables only (no parent dbids), or a mixture of the two, depending on in request parameters.

udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GrantedDBs

<qdbapi>
   <ticket>auth_ticket</ticket>
   <udata>mydata</udata>
   <includeancestors>1</includeancestors>
   <excludeparents>1</excludeparents>
</qdbapi>
```

URL alternative

```
https://target_domain/db/main?a=API_GrantedDBs
&ticket=auth_ticket&excludeparents=1&includeancestors=1
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_GrantedDBs</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>mydata</udata>
   <databases>
      <dbinfo>
         <dbname>AppTokenCreateTest</dbname>
         <dbid>bdzk2ecg5</dbid>
         <ancestorappid>beaa6db7t</ancestorappid>
         <oldestancestorappid>bd9jbshim</oldestancestorappid>
      </dbinfo>
      <dbinfo>
         <dbname>TestTable1</dbname>
         <dbid>bdzuh4nj5</dbid>
      </dbinfo>
   </databases>
</qdbapi>
```

Did this help you? Give us a rating:

API_GrantedDBsForGroup

Overview

Use API_GrantedDBsForGroup to get a list of all the applications that a group can access.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the group to manage.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_GrantedDBsForGroup.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
databases	A list of information about the databases where this group has access.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION:API_GrantedDBsForGroup

<qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <gid>1217.dgpt</gid>
</qdbapi>
```

Sample XML Get Request

```
https://target_domain/db/main?a=API_GrantedDBsForGroup&gid=1217.dgpt&ticket=auth_ticket&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<qdbapi>
<action>API_GrantedDBsForGroup</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<databases>
<dbinfo>
<dbname>Projects</dbname>
<dbid>bhgnyxp3v</dbid>
</dbinfo>
<dbinfo>
<dbname>Project Manager Plus</dbname>
<dbid>bhgnyxp3u</dbid>
</dbinfo>
</databases>
</qdbapi>
```

Did this help you? Give us a rating:

API_GrantedGroups

Overview

Use API_GrantedGroups to get the list of groups to which a user has been granted access. If the adminOnly flag is passed, then only return groups in which this user has administration privileges.

Request parameters

PARAMETER	VALUE	REQUIRED?
userid	The user id.	yes
adminonly	Only return groups where the user has admin rights. Default is FALSE.	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_GrantedGroups.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
groups	A list of the user's groups.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_GrantedGroups

<qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <userid>930245.jlpw</userid>
</qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/main?a=API_GrantedGroups&userid=930245.jlpw&
ticket=auth_ticket&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_GrantedGroups</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
    <groups>
       <group id=”1217.dgpt”>
          <name>GroupInfoTestGroup</name>
          <description>Demo Test Group</description>
          <managedByUser>false</managedByUser>
       </group>
       <group id=”1218.sfdl”>
          <name>AnotherGroupInfoTestGroup</name>
          <description>Yet Another Demo Test Group</description>
          <managedByUser>true</managedByUser>
       </group>
    </groups>
</qdbapi>
```

Did this help you? Give us a rating:

API_ImportFromCSV

Overview

Use API_ImportFromCSV to add or update several records. You invoke this call on a table-level dbid. You can add AND update in the same API_ImportFromCSV request. (For an Add, leave the Record ID empty.)

The **clist** parameter is optional when adding new records to a table. When updating existing records, the **clist** parameter must contain the field ID for the key field (usually *3* for the *Record ID#*). The CSV file you're importing must also include a column that contains the value of the key field for each record that will be updated.

Request parameters

PARAMETER	VALUE	REQUIRED?

records_csv	An aggregate containing the actual records you are importing.
	`<records_csv>`
	`    <![CDATA[`
	`        your, record, values, go, here`
	`        and, hopefully, here, as, well`
	`    ]]>`
	` </records_csv>`
	For field format, usage, and any validation for fields of particular types, see API_AddRecord.
		yes

PARAMETER	VALUE	REQUIRED?
clist	A period-delimited list of field IDs that indicates how the columns in the CSV file will map to fields in the table when the import happens. To prevent a column in the CSV file from being imported, enter a 0 in the field ID list. This means that the first field ID in the list maps to the first column in the CSV file, the second field ID maps to the second column in the CSV file, and so forth. In the following example, the CSV file contains 4 columns. Quick Base will not import either the first or the third columns. The second column will map to field ID 7, and the fourth column will map to field ID 6. `<clist>0.7.0.6</clist>`	Yes, if you are updating records, or you want the columns in the CSV file to map to certain fields in the table.
clist_output	Specifies which fields should be returned in addition to the record ID and updated ID. Specify a period-delimited clist string, for example: `<clist_output>0.7.0.6</clist_output>`	no
decimalPercent	Set this parameter to 1 so that decimal values like 0.50 will be interpreted to mean 50%.	no
skipfirst	Set this parameter to 1 to prevent Quick Base from importing the first row of data in a CSV file. You must specify this parameter if the first row of your CSV file contains column names. Omit this property if you don't want to prevent the first row to be imported.	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no
msInUTC	Allows you to specify that Quick Base should interpret all date/time stamps passed in as milliseconds using Coordinated Universal Time (UTC) rather than using the local application time. Set this parameter to 1 if you want to use Coordinated Universal Time. See usage example.	no

PARAMETER	VALUE	REQUIRED?
mergeFieldId	With this parameter, the import uses the field specified as the key, instead of the key field for the table. • The merge field must be a unique field, and the specified merge field must contain unique values. • Values for the table key and/or record ID field will be ignored from input to prevent data corruption. • If the specified merge field uses a formula, and a record cannot be found by its value, a new record will be added if the resulting formula value is a valid value for the field. • Record ID or other table key field can be used as the specified merge field. • The clist parameter is required if this parameter is used. To specify the field, use the field ID, for example: `<mergeFieldId>10</mergeFieldId>` For more details, read an article about the parameter on the Quick Base Community site.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_ImportFromCSV.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.

ELEMENT NAME	VALUE
udata	Optional. Contains any udata value supplied in the request.
num_recs_added	The number of records added to the table
num_recs_input	The total number of records in the CSV file
num_recs_updated	The number of records in the table that were updated
rids	A list of the record IDs for all the records that were either added or updated
update_id	Attribute of each rid element that defines the update_id for the record. Update IDs are used to detect update conflicts in subsequent operations with API_EditRecord.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_ImportFromCSV

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <records_csv>
      <![CDATA First Name,Last Name,Company,Phone,Cell Phone,Zip
         Bruce,Anderson,Reyes Inc,(474) 555-0514,(390) 555-8927,<-80145>
         Judy,Atwell,Conner Supplies,(499) 555-1072,(763) 555-1325,<-50737>
         Kris,Babs,Willis Orchards,(428) 555-6791,(481) 555-1335,<-81504>]]>
   </records_csv>
   <clist>7.8.6.5.4</clist>
   <skipfirst>1</skipfirst>
</qdbapi>
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample XML Request to Add New Records

```
<qdbapi>
<records_csv>
<![CDATA[First Name,Last Name,Company,Phone,Cell Phone,Zip
Bruce,Anderson,Reyes Properties,(474) 555-0514,(390) 555-8927,<-80145>
Judy,Atwell,Conner Garden Supplies,(499) 555-1072,(763) 555-1325,<-50737>
Kristie,Babcock,Willis Orchards,(428) 555-6791,(481) 555-1335,<-81504>
Linda,Bailey,Willis Orchards,(544) 555-8912,(491) 555-1449,<-07738>
Mike,Balak,Nye Properties,(253) 555-9766,(637) 555-4566,<-93341>
Renee,Barley,Lew Plumbing,(486) 555-4747,(268) 555-5904,<-98841>
Howard,Bayne,Holly Heating and Electric,(743) 555-2294,(756) 555-7934,<-40984>
David Y.,Becker,Express Service,(825) 555-8433,(255) 555-2867,<-49873>
]]></records_csv>
<clist>7.8.6</clist>
<skipfirst>1</skipfirst>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
</qdbapi>
```

Sample XML Response (Add New Records)

```
<?xml version="1.0" ?>
<qdbapi>
 <action>API_ImportFromCSV</action>
 <errcode>0</errcode>
 <errtext>No error</errtext>
 <num_recs_input>8</num_recs_input>
 <num_recs_added>8</num_recs_added>
<rids>
 <rid update_id="1057961999003">1</rid>
 <rid update_id="1057961999003">2</rid>
 <rid update_id="1057961999003">3</rid>
 <rid update_id="1057961999003">4</rid>
 <rid update_id="1057961999003">5</rid>
 <rid update_id="1057961999003">6</rid>
 <rid update_id="1057961999003">7</rid>
 <rid update_id="1057961999003">8</rid>
</rids>
</qdbapi>
```

Sample Response (Update Existing Records)

```
<qdbapi>
 <action>API_ImportFromCSV</action>
 <errcode>0</errcode>
 <errtext>No error</errtext>
 <num_recs_input>2</num_recs_input>
 <num_recs_updated>2</num_recs_updated>
 <rids>
 <rid>7</rid>
 <rid>8</rid>
 </rids>
</qdbapi>
```

Did this help you? Give us a rating:

API_ProvisionUser

Overview

Use API_ProvisionUser to add a user who is not yet registered with Quick Base to your application. You invoke an application-level dbid **for a user that is not yet registered with Quick Base**, but whose email is known to you. This call:

- starts a new user registration in Quick Base using the supplied email, fname (first name), and lname (last name).
- gives application access to the user by adding the user to the specified role.

Note that, to assign roles to users, you must have either of these permissions on the application:

- **Basic Access with Sharing** -- If you have Basic Access with Sharing permissions, you can assign roles with all permissions except Full Administration.
- **Full Administration** -- If you have Full Administration permissions, you can assign roles with all permissions.

After you invoke this call, you'll need to invoke API_SendInvitation to invite the new user via email. When the user clicks on the email invitation, the user is prompted to complete the brief registration. (At this time, the user can change the first and last name you assigned.)

If a user already is registered with Quick Base, the API will return an error. With registered users, you should use API_GetUserInfo, API_AddUserToRole, and API_SendInvitation.

Request parameters

PARAMETER	VALUE	REQUIRED?
email	The email address of the person to whom you are granting access.	yes
roleid	The role ID of the role you want to assign this user to. You can find this information by invoking API_GetRoleInfo. If you don't supply a roleid, the role will be set to *none*.	no

PARAMETER	VALUE	REQUIRED?
fname	The first name of the new Quick Base user.	yes
lname	The last name of the new Quick Base user.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_ProvisionUser.

ELEMENT NAME	VALUE
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
userid	The userid for the new user returned by Quick Base.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_ProvisionUser

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <roleid>11</roleid>
   <email>sanskor@sbcglobal.com</email>
   <fname>Margi</fname>
   <lname>Rita</lname>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_ProvisionUser&email=
DukeOHazard@sbcglobal.net&roleid=11&fname=Muggsy&lname=Bogues&ticket=
auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>api_provisionuser</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <userid>112248.5nzg</userid>
</qdbapi>
```

Did this help you? Give us a rating:

API_PurgeRecords

Overview

Use API_PurgeRecords to delete the specified records from the specified table. All records matching the criteria in the specified query will be deleted.

You can build your own query string using the **query** parameter in the call. You can also use a saved query for the table, using **qid** or **qname**. See API_DoQuery for information on saved queries or for instructions on building the query string.

Caution: Use this call carefully. If you do not supply a query parameter (query or qid or qname), ALL of the table records will be deleted. If you supply an empty query parameter (<query />, or <qid /> or <qname />) all of the table records will be deleted as well. If you need to delete only one record, you should use API_DeleteRecord.

Request parameters

PARAMETER	VALUE	REQUIRED?
query or qid or qname	Specifies the query. You can use any of these three options: • Use `<query>{myfid.operator.value}</query>` to build your own query string. • Use `<qid>n</qid>`if you want to use a saved query and want to use its query ID. • Use `<qname>myQuery</qname>` if you want to use a saved query and refer to it by its query name.	No, but unless you supply a non-empty query, all records in the table are deleted
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	Supply a valid application token.	yes, if the application requires application tokens.
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_PurgeRecords.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
num_records_deleted	The total number of records that were deleted.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_PurgeRecords

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <qid>10</qid>
</qdbapi>
```

URL Alternative

```
https://target_domain/db/target_dbid?a=API_PurgeRecords&qid=10
&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_PurgeRecords</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <num_records_deleted>21</num_records_deleted>
</qdbapi>
```

Did this help you? Give us a rating:

API_RemoveGroupFromRole

- Overview
- Request parameters
- Response values
- Sample XML Request
- Sample HTTP Get Request
- Sample response

Overview

Use API_RemoveGroupFromRole to remove a group from a role in an app. You can also indicate if you want to remove a group from all roles in an application.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the group to manage.	yes
roleid	The ID of the access role being assigned to the group.	yes
apptoken	A valid application token.	yes, if the application requires application tokens
allRoles	Set to TRUE if the group is to be removed from all roles in the app. Default is FALSE.	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_RemoveGroupFromRole.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_RemoveGroupFromRole
```

```
<qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <gid>345889.klsd</gid>
   <roleid>12</roleid>
   <allRoles>true</allRoles>
</qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/target_dbid?a=API_RemoveGroupFromRole&gid=345889.klsd
&roleid=12&allRoles=true&ticket=auth_ticket&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_RemoveGroupFromRole</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_RemoveSubgroup

- Overview
- Request parameters
- Response values
- Sample XML Request
- Sample HTTP Get Request
- Sample response

Overview

Use API_RemoveSubgroup to remove a nested group from a group.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the containing group.	yes
subgroupid	The id of the subgroup.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_RemoveSubgroup.

ELEMENT NAME	VALUE
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_RemoveSubgroup

<qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <gid>345889.jksd</gid>
   <subgroupid>345889.skld</subgroupid>
</qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/main?a=API_RemoveSubgroup&gid=345889.jksd
&subgroupid=345889.skld&ticket=auth_ticket&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_RemoveSubgroup</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_RemoveUserFromGroup

- Overview
- Request parameters
- Response values
- Sample XML Request
- Sample HTTP Get Request
- Sample response

Overview

Use API_RemoveUserFromGroup to remove a user from a group.

Request parameters

PARAMETER	VALUE	REQUIRED?
gid	The id of the group to manage.	yes
uid	The id of the user to be removed from the group.	yes
email	The email of the Quick Base user	no
screenName	The screen name of the Quick Base user, if set	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_RemoveUserFromGroup.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.1
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION:API_RemoveUserFromGroup

<qdbapi>
   <udata>misc data</udata>
   <ticket>auth_ticket</ticket>
   <gid>345889.sjkl</gid>
   <uid>9380434.rtgf</uid>
</qdbapi>
```

Sample HTTP Get Request

```
https://target_domain/db/main?a=API_RemoveUserFromGroup&gid=345889.sjkl
&userid=9380434.rtgf&ticket=auth_ticket&udata=misc%20data
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_RemoveUserFromGroup</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_RemoveUserFromRole

- Overview
- Request parameters
- Response values
- Sample XML Request
- URL alternative
- Sample response

Overview

Use API_RemoveUserFromRole to remove a user from a specified role. You invoke this call on an application-level dbid. If the user has no other role, this call eliminates the user from the application's role list. You'll need to get the user ID by calling API_GetUserInfo if you want to assign the user to another role in the future.

This call can be used to remove the user entirely from any role in the application, effectively turning off access to that user. If you intend to turn off all access, you would need to call API_GetUserRole Info to see what roles the user has, then invoke API_RemoveUserFromRole on each role.

If you expect to add that user to another role in the future, you should consider using API_ChangeUserRole, which can be used to turn off access (with a role set to *None*) while keeping the user on the application's role list for future reinstatement or role change.

If you are simply changing the user from one role to another, you should use API_ChangeUserRole.

Request parameters

PARAMETER	VALUE	REQUIRED?
userid	The ID of user you want removed from the role.	yes
roleid	The ID of the role from which you want the user removed.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: - ticket - username/password - user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_RemoveUserFromRole.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
num_records_deleted	The total number of records that were deleted.

Sample XML Request

```
POST https://target_domain/db/target_dbid
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION:API_RemoveUserFromRole

<qdbapi>
   <udata>misc data</udata>
   <userid>112245.efy7</userid>
   <roleid>11</roleid>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_RemoveUserFromRole&userid=112245
&roleid=11&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_RemoveUserFromRole</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>misc data</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_RenameApp

Overview

Use API_RenameApp to change an application's name. You invoke this call on an application-level dbid. You must have full administration rights on the application to use this call.

Request parameters

PARAMETER	VALUE	REQUIRED?
newappname	The name you want to assign to the application	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_RenameApp.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_RenameApp

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <newappname>Refueler</newappname>
   <udata>mydata</udata>
</qdbapi>
```

URL Alternative

```
https://target_domain/db/target_dbid?a=API_RenameApp&newappname=Refueler
&ticket=auth_ticket&apptoken=app_token
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_RenameApp</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>mydata</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_RunImport

Overview

Quick Base allows you to import data from one or more of your application tables into another application table. You can import tables from within your application, or from another application. (For complete information on this feature, see the Quick Base online help.)

If you'll need to execute the same import multiple times, you can save a specific import and execute it any time you need to run it. To run a saved import, invoke API_RunImport on a **table dbid** whose ID you specify in this call. Note that you can’t use API_GetSchema to get the saved import's ID because API_GetSchema doesn’t return the saved import.

To access the saved import ID:

1. Open the application into which you want to import data.
2. On the app Home page, click **Import/Export** on the Page bar.
3. Select the **Import into a table from another table** radio button.
4. Click **Import from Another Table**.
5. Click the name of the saved import you want to use. In the URL shown in the browser you will see this parameter: **&id=X**. *X* is the ID you need to supply to this call.

Request parameters

PARAMETER	VALUE	REQUIRED?
id	The ID of the saved import that you want to execute.	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_RunImport.
errcode	If successful, this will be 0. If there is a failure, there will be a code returned here. See Appendix A for a list of codes.
errtext	If successful, this will be “No error”. If there is a failure this contains more detail on the nature of the failure.
import_status	Describes the result of the import you just executed, for example “3 new records were created”.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_RunImport

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <id>10</id>
</qdbapi>
```

URL alternative

```
https://target_domain/db/target_dbid?a=API_RunImport&id=10
&apptoken=app_token&ticket=auth_ticket
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_RunImport</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <import_status>3 new records were created.</import_status>
</qdbapi>
```

Did this help you? Give us a rating:

API_SendInvitation

Overview

Use API_SendInvitation to send an email invitation to your application. You can send an email invitation to either an existing Quick Base user that you have granted application access to via API_AddUserToRole, or to a new Quick Base user that you have created via API_ProvisionUser.

Request parameters

PARAMETER	VALUE	REQUIRED?
userid	The ID of the Quick Base user you are inviting to your application.	yes
usertext	The message you want to display in your email invitation.	no
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	Echoes the originating request, for example, API_SendInvitation.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_SendInvitation

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <userid>112249.ctdg</userid>
   <usertext>Welcome!</usertext>
</qdbapi>
```

URL alternative

`https://`*target_domain*`/db/`*target_dbid*`?a=API_SendInvitation&userid=112249.ctdg&ticket=`*auth_ticket*`&apptoken=`*app_token*`&usertext=Hello`

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_SendInvitation</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
</qdbapi>
```

Did this help you? Give us a rating:

API_SetDBvar

Overview

Use API_SetDBVar to create a database variable (DBVar) or set a value for it. If the DBVar already exists, this call overwrites the existing value.

You invoke this call on an application-level id. You can invoke this call on one DBVar at a time. You must have full administration rights on the application to use this call.

Request parameters

PARAMETER	VALUE	REQUIRED?
varname	The name you want the DBVar to have.	yes
value	The value you want to set in the DBVar	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens

PARAMETER	VALUE	REQUIRED?
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_SetDBVar.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_SetDBVar

<qdbapi>
   <udata>mydata</udata>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <varname>usercode</varname>
   <value>14</value>
</qdbapi>
```

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_SetDBVar</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
</qdbapi>
```

Did this help you? Give us a rating:

API_SetFieldProperties

Overview

Use API_SetFieldProperties to set one or more properties of a field. You invoke this on a table-level dbid.

Typically, you use this call after you create a new field using API_AddField, to set up its default behavior. You can also use this call at any time if you want to change properties, even if the affected field has data.

The properties available for a field vary slightly for the different field types. To get all of the available properties for a field, and to get the field id (fid) needed, use the API_GetSchema call.

Request Parameters

PARAMETER	VALUE	REQUIRED?
<property name>property value</property name> For example: `<default_value>Hello</default_value>`	Name/value pairs for the properties to be set.	yes
fid	Field ID of the field to be changed	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token

PARAMETER	VALUE	REQUIRED?
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Properties for all field types

PROPERTY TAG	DESCRIPTION	VALID FOR THESE FIELD TYPES
appears_by_default	Specifies whether this field column automatically appears in all reports (views) for this table or whether this field must be explicitly added to each one. Acceptable values are: • 1 - all views have this field by default. • 0 - no views have this field by default	All

PROPERTY TAG	DESCRIPTION	VALID FOR THESE FIELD TYPES
bold	Specifies whether the value should display in bold font. Acceptable values are: • 1 - turn on bolding • 0 - turn off bolding (default)	All
comments	Specifies an entry for the Comments on a field. Use this to enter free-form notes about field usage. Limit of 500 characters.	
default_value	The text or value to be displayed in the field by default. Specify the desired value.	All
fieldhelp	Lets you specify help text for the field. This results in a help icon (?) being displayed for the field in the Add record form. The content of fieldhelp is shown as a tooltip for that icon and displays if you double-click the icon. Enter the desired help text.	All
find_enabled	Specifies whether Quick Base should search on this field. For more information on how this property works, see the Quick Base online help. • 1 - Quick Base can search on this field. (default) • 0 - Quick Base will not search on this field.	All
label	The text that shows up as the column header in the table. Any text value, limited by the **maxlength** and **num_lines** properties.	All

PROPERTY TAG	DESCRIPTION	VALID FOR THESE FIELD TYPES
no_wrap	Allows/disallows wrapping of the displayed value. Acceptable values are: • 1 - disallows wrap (default) • 0 - allows wrap	All
num_lines	Specifies the number of lines in a Text field.	Text field types: • Text • Text - Multi-line
required	Specifies whether a valid field value must be supplied or is optional. Acceptable values are: • 1 - value is required • 0 - value is optional (default)	All
unique	Specifies whether the field value must be unique within the table. Acceptable values are: • 1 - must be unique • 0 - duplicate values are allowed (default)	All
width	Specifies the width in characters of the column in views. Specify the desired number of characters.	All

PROPERTY TAG	DESCRIPTION	VALID FOR THESE FIELD TYPES
doesdatacopy	In the Quick Base UI, after you add a new record, the resulting display form shows a button called **Add Similar**. If the user clicks **Add Similar** to add another record, all fields with this property set to 1 will be prefilled with values taken from that previously added record, overriding the **default_value** property.	All field types that accept user input
append_only	Allows you to keep a record of all entries for a field in a running log. Append fields are regular text fields which record each entry and stamp the text with the name of the user who made the change and the date on which they did it. All entries display when the field is viewed or edited The running log includes one line for the entry and the username of the person who originated the entry. New entries are appended at the end. You can turn this on or off after you make appended entries: the entries already made will be retained even if the feature is turned off. If you want an expanded style to be used, or time of day stamp, or the new entries to be prepended, you will have to change these settings in the Quick Base UI. Acceptable values are: • 1 - is an append field • 0 - is NOT an append field (default)	Text field types: • Text • Text - Multi-line • Text - Multiple-choice
maxlength	Specifies the maximum number of characters allowed in this field. Acceptable values are: • 0 - no limit • Any number other than 0.	Text field types: • Text • Text - Multi-line • Text - Multiple Choice

PROPERTY TAG	DESCRIPTION	VALID FOR THESE FIELD TYPES
allowHTML	Enables HTML code within a rich text field. If this property is turned on users can enter HTML tags to format text, create custom links while in edit mode. When the record displays, Quick Base renders the code just like any browser would.	Rich Text
choices	Aggregates default choices in a multiple-choice or Multi-select Text field, in this way: `<choices>    <choice>1</choice>    <choice>2</choice>  <choice>N</choice> </choices>` Specify the aggregate and choices. **Note:** Choices added to a Multi-select Text field are limited to 60 characters, and the total number of choices in the field may not exceed 100.	All Numeric field types Text Text - Multi-line Text - Multiple Choice Multi-select Text
sort_as_given	Specifies how the sorting is to be done on the multiple choices in this field: alphabetically or following the order in which the choices were created. Acceptable values are: • 1 - sort in the order of creation • 0 - sort alphabetically (default)	All Numeric field types Text Text - Multi-line Text - Multiple Choice Multi-select Text

PROPERTY TAG	DESCRIPTION	VALID FOR THESE FIELD TYPES
allow_new_choices	Specifies whether users can add their own choices to Text - Multiple Choice and Multi-select Text fields, or whether they are limited to accepting one of the displayed values. - 1 - allow new choices - 0 - disallow new choices (default) **Note:** Choices added to a Multi-select Text field are limited to 60 characters, and the total number of choices in the field may not exceed 100.	All Numeric field types Text Text - Multi-line Text - Multiple Choice Multi-select Text User
blank_is_zero	Specifies whether blank entries in this field will be treated as though the value "0" was entered. Acceptable values are: - 1 - treat blank entries as zero (default) - 0 - do not treat blank entries as zero	Duration All Numeric field types
numberfmt	Specifies the current number format (digit grouping, thousands separator, and decimal separator). Acceptable values are: - 0 - 12345678.00 - 3 - 12,345,678.00 - 6 - 12.345.678,00 - 7 - 12345678,00 - 8 - 1,23,45,678.00 - 10 - 1.23.45.678,00	All Numeric field types

PROPERTY TAG	DESCRIPTION	VALID FOR THESE FIELD TYPES
comma_start	Specifies whether and how commas are to be used in numeric displays. Acceptable values are: • 3 - commas should be used in numbers larger than 999 • 4 - commas should be used in numbers larger than 9999	All Numeric field types
does_average	Specifies whether Quick Base computes the average of all the values in this field and displays the result at the bottom of the view (AVG row). Acceptable values are: • 1 - display average • 0 - do not display average	Duration All Numeric field types
does_total	Specifies whether Quick Base should compute the sum of all the values in this field and display the result at the bottom of the view (TOT row). Acceptable values are: • 1 - calculate totals • 0 - do not calculate totals	Duration Checkbox All Numeric field types
currency_format	Specifies the placement of the currency symbol. Acceptable values are: • 0 - before the number ($-1.00) • 1 - between the "-" and the number (-$1.00, default) • 2 - after the number (-1.00$)	Numeric - Currency

PROPERTY TAG	DESCRIPTION	VALID FOR THESE FIELD TYPES
currency_symbol	Specifies the currency symbols to be displayed in this field. The default currency symbol is the dollar sign ($). You can enter a different currency symbol if you'd like. You can enter the currency symbol as either an HTML character entity or as a Unicode character code. For example, to display a pound sign (£), you would enter one of the following in the **Symbol** box: - &pound; (HTML character entity) - 0163 while holding down the ALT key (Unicode character code)	Numeric - Currency
decimal_places	Specifies the number of digits you want to display after the decimal point. For example, you might enter "2" if the field is a price field. If a user enters more digits after the decimal point than you've entered here, Quick Base rounds the number and extends it only to the number of decimal places you specified. The fraction .5 rounds the number away from 0. For example, 3.5 rounds to 4, and -3.5 rounds to -4. If you would rather .5 always round up, meaning that -3.5 would round to -3, use the Round formula instead of the decimal places option for rounding. Enter the number of digits you want to display after the decimal point. Leave blank for floating point.	- All Numeric field types - Duration - Time of Day - Formula - Numeric - Formula - Duration - Formula - Time of Day

PROPERTY TAG	DESCRIPTION	VALID FOR THESE FIELD TYPES
default_today	Specifies whether the default date is today's date. Acceptable values are: • 0 - no default date (default) • 1 - make the default date today's date	Date Date/Time Formula fields of those types
display_dow	Turns the day of the week display on or off. For example, if you turn this feature now, July 10, 2010 displays as follows: Saturday, JUL 10 (with display_month set to name and display_relative set to 1) Acceptable values are: • 0 - do not show day of week (default) • 1 - show day of week	Date Date/Time Formula fields of those types
display_month	Specifies whether the month is displayed as a name or a number. For example, July as a number is displayed as 07. As a name it is displayed as JUL. Acceptable values are: • number - display month as a number from 1 to 12 • name - display the common three-character abbreviated name for the month	Date Date/Time Formula fields of those types

PROPERTY TAG	DESCRIPTION	VALID FOR THESE FIELD TYPES
display_relative	Turns the Quick Base smart date feature on or off. This feature changes the display based on relative date/times. For example, a date that is today is displayed simply as “Today”. A date that falls within the current year omits the year, and so forth. See the Quick Base online help for full description of this feature. Acceptable values are: • 0 - turn smart date off (default) • 1 - turn smart date on	Date Date/Time Formula fields of those types
display_time	Specifies whether the field shows date only in Quick Base UI or both date and time. This option turns a date field into a date-time field. If you want both date and time displayed, you must supply both a date and a time or a default time will be supplied. You can turn off the time display whenever you want. However any times already supplied will be saved and will reappear if you turn time display back on. Acceptable values are: • 0 - date-only display (default) • 1 - display date and time	Date Date/Time Formula fields of those types
display_zone	Turns on the display of time zone in certain hidden fields, such as time created and time last modified. Acceptable values are: • 0 - turn off time zone display (default) • 1 - turn on time zone display	Date/Time Formula - Date/Time
has_extension	Used in phone field to specify whether an extension box will be displayed to the right of the telephone number. Acceptable values are: • 1 - show the extension box (default) • 0 - hide the extension box	Phone Number

PROPERTY TAG	DESCRIPTION	VALID FOR THESE FIELD TYPES
display_as_button	Specifies whether a URL should display as a link or a button. Acceptable values are: • 0 - as a link (default) • 1 - as a button (instead of as a link)	URL Formula - URL
formula	Specifies a valid formula for a formula field.	All Formula fields

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_SetFieldProperties.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.
fid	The id of the field whose properties were successfully set.
fname	The label of the field whose properties were set.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0Content-Type:
application/xmlContent-Length:QUICKBASE-ACTION: API_SetFieldProperties

<qdbapi>   <ticket>auth_ticket</ticket>   <apptoken>app_token</apptoken>   <udata>mydata</udata>   <fid>6</fid>   <label>Business Phone
Number</label>   <nowrap>0</nowrap>   <bold>0</bold>   <required>1</required>
<display_as_button>0</display_as_button>   <appears_by_default>1</appears_by_default>   <find_enabled>1</find_enabled>   <allow_new_choices>0</allow_new_choices>   <sort_as_given>0</sort_as_given>   <carrychoices>0</carrychoices>   <foreignkey>0</foreignkey>   <unique>1</unique>   <doesdatacopy>0</doesdatacopy>   <fieldhelp>Contact phone
number</fieldhelp>   <num_lines>1</num_lines>   <append_only>0</append_only>   <allowHTML>1</allowHTML>   <has_extension>1</has_extension></qdbapi>
```

URL alternative

`https://`*target_domain*`/db/`*target_dbid*`?a=API_SetFieldProperties&fid=7&default_value=GoodBye&ticket=`*auth_ticket*`&apptoken=`*app_token*

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

To set choices for a Text - Multiple Choice or Multi-select Text field using a URL alternative, use API_FieldAddChoices instead.

Sample response

```
<?xml version="1.0"
?><qdbapi><action>API_SetFieldProperties</action>   <errcode>0</errcode>   <errtext>No error</errtext>   <udata>mydata</udata>   <fid>6</fid>   <fname>Business
Phone Number</fname></qdbapi>
```

Did this help you? Give us a rating:

API_SetKeyField

Overview

Use API_SetKeyField to specify a field in the table as the key field. You invoke this call on a table dbid. You must have full administration rights on the application to use this call.

If you don't specify a key field, Quick Base uses the built-in Record ID field as the key field. However, if you want to use a different field as the key field for a table, the following conditions must be met:

- The field type you pick must support uniqueness (make sure the Unique checkbox is checked, which disallows duplicate entries).
- If there is data in the table, the field values must all be unique and cannot be blank.
- You cannot specify a List - user or Multi-select text field, or any formula field, as the key field in a table.

Request parameters

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens

PARAMETER	VALUE	REQUIRED?
fid	The field ID of the table field to be used as the key field.	yes
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_SetKeyField.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid?
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_SetKeyField

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
   <fid>6</fid>
   <udata>myudata</udata>
</qdbapi>
```

URL alternative

`https://`*target_domain*`/db/`*target_dbid*`?a=API_SetKeyField&fid=7`
`&ticket=`*auth_ticket*`&apptoken=`*app_token*

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_SetKeyField</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <udata>myudata</udata>
</qdbapi>
```

Did this help you? Give us a rating:

API_SignOut

Overview

This call is for use by API client implementations that make use of the ticket cookie rather than the <ticket> parameter.

Invoking this call returns a null ticket cookie (with the name TICKET). In some cases, invoking API_SignOut results in applications at the local machine (the API client) being unable to access Quick Base applications until API_Authenticate is called for a new ticket cookie.

This call does not invalidate any tickets, nor log off the caller from any Quick Base applications, nor prevent further access of Quick Base applications. If the caller has saved a valid ticket, that caller can continue to use that ticket even after API_SignOut is called.

Request parameters

PARAMETER	VALUE	REQUIRED?
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_SignOut
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.

ELEMENT NAME	VALUE
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/main HTTP/1.0
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_SignOut

<qdbapi>
</qdbapi>
```

URL alternative

```
https://target_domain/db/main?a=API_SignOut
```

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>api_signout</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
</qdbapi>
```

Did this help you? Give us a rating:

API_UploadFile

Overview

Use API_UploadFile to upload base64-encoded file attachments in Quick Base.

Request parameters

PARAMETER	VALUE	REQUIRED?

field	Specify values for file attachment fields using either the **fid** attribute or **name** attribute of the <field> element.
	In the opening <field> tag, insert the **filename** attribute in addition to the **fid** or **name** attribute.
	The **filename** attribute value should be set to the name of the file with no path specified. Insert the base64-encoded text from the encoded file between the opening and closing <field> tags.
	For example:
	`<field fid="18" filename="photo1.jpg">…</field>`
	`<field name="Message" filename="photo2.jpg"> …</field>`
	The file attachment must be a base64-encoded file. See Managing Files for more information about uploading files.
	Note that you must not use MIME encoding and must not include MIME headers. Many base64 encoders or base64 encoding methods are for MIME type encoding and will not work with Quick Base.
	You must not insert any new line characters when you base64 encode the file. If your file attachments appear to upload but don’t display in Quick Base, double check for presence of these characters.
		yes

PARAMETER	VALUE	REQUIRED?
rid	The id of the record that will contain the file attachment	yes
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token	yes, if the application requires application tokens

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_UploadFile.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.

ELEMENT NAME	VALUE
file_fields	Contains a list of <field> elements for the uploaded file(s). Each <field> element includes a <url> element that can be used to fetch a specific uploaded file.

Sample XML Request

See XML Request example with a truncated base64-encoded file attachment.

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
   <action>API_UploadFile</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <file_fields>
      <field id="13"> <!--(Uploaded File Name)-->
         <url>https://target_domain/up/bc4gzy4nx/g/rc/ep/va/qchain.log</url>
      </field>
   </file_fields>
</qdbapi>
```

Did this help you? Give us a rating:

API_UserRoles

Overview

Use API_UserRoles to get details about an application's users and their roles. This call returns all users and their roles.

Note: In order to use the API_UserRoles call, you must have either Basic Access with Sharing access or Full Administration access to the application. If you have only Basic Access with Sharing access, you cannot add a user to a role that has Full Administration permissions.

Request parameters

PARAMETER	VALUE	REQUIRED?
ticket	A valid authentication ticket. The authentication ticket is returned via the API_Authenticate call.	yes, one of: • ticket • username/password • user token
usertoken	The user token is an alternative means of authentication, used for API access. User tokens cannot be used to access the Quick Base UI.	yes, one of: • ticket • username/password • user token
apptoken	A valid application token.	yes, if the application requires application tokens
udata	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response.	no

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_AddField.
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
users	This aggregate contains one or more user sub-aggregates. Each of the user sub-aggregates contains the actual data about a user or group that has been assigned a role in the app. The userid or groupid is returned as an attribute in the user sub-aggregate. The user sub-aggregrate also contains the name and role information, structured as shown in the Sample Response section below. Each role returned has a role ID as an attribute and contains a role name and the application access level that is conferred by the role.
udata	Optional. Contains any udata value supplied in the request.

Sample XML Request

```
POST https://target_domain/db/target_dbid HTTP/1.0
Content-Type: application/xml
Content-Length: 77
QUICKBASE-ACTION: API_UserRoles

<qdbapi>
   <ticket>auth_ticket</ticket>
   <apptoken>app_token</apptoken>
</qdbapi>
```

URL alternative

`https://`*target_domain*`/db/`*target_dbid*`?a=API_UserRoles`
`&apptoken=`*app_token*

where *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Sample response

```
<?xml version="1.0" ?>
 <qdbapi>
   <action>API_UserRoles</action>
   <errcode>0</errcode>
   <errtext>No error</errtext>
   <users>
      <user type="user" id="112149.bhsv">
         <name>Jack Danielsson</name>
         <lastAccess>1403035235243</lastAccess>
         <lastAccessAppLocal>06-17-2014 01:00 PM</lastAccessAppLocal>
         <firstName>Jack</firstName>
         <lastName>Danielsson</lastName>
         <roles>
            <role id="12">
               <name>Administrator</name>
               <access id="1">Administrator</access>
            </role>
         </roles>
      </user>
      <user type="user" id="112245.efy7">
         <name>Ivar Boneless</name>
         <lastAccess/>
         <lastAccessAppLocal/>
         <firstName>Ivar</firstName>
         <lastName>Boneless</lastName>
         <roles>
            <role id="10">
               <name>Viewer</name>
               <access id="3">Basic Access</access>
            </role>
         </roles>
      </user>
      <user type="group" id="3">
        <name>Everyone on the Internet</name>
        <roles>
         <role id="10">
            <name>Viewer</name>
            <access id="3">Basic Access</access>
          </role>
        </roles>
      </user>
   </users>
</qdbapi>
```

Did this help you? Give us a rating:

API_Webhooks_Activate

Overview

Use API_Webhooks_Activate to make a webhook active.

Request Parameters

PARAMETER	VALUE	REQUIRED?
actionIDList	Comma-separated IDs of the webhook	yes

Response values

PROPERTY TAG	DESCRIPTION
numChanged	Denotes how many webhooks were activated

Sample XML Request

```
POST https://target_domain/db/target_dbid?
 Content-Type: application/xml
 Content-Length:
 QUICKBASE-ACTION: API_Webhooks_Activate
 <qdbapi>
 <usertoken>buacrj_uyp_iag2s56b7t8opp5edsvt</usertoken>
 <actionIDList>1</actionIDList>
 </qdbapi>
```

URL alternative

```
https://myApp.quickbase.com/db/bmdihpya6?a=API_Webhooks_Activate&actionIDList=1
```

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
<action>API_Webhooks_Activate</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<numChanged>1</numChanged>
<success>true</success>
</qdbapi>
```

Did this help you? Give us a rating:

API_Webhooks_Copy

Overview

Use API_Webhooks_Copy to make a duplicate of an existing webhook.

Request Parameters

PARAMETER	VALUE	REQUIRED?
actionID	ID of the action to copy	yes

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_Webhooks_Copy
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 (zero) indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
actionID	actionID of the webhook

ELEMENT NAME	VALUE
success	true or false

Sample XML Request

```
POST https://target_domain/db/target_dbid?
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_Webhooks_Copy
<qdbapi>
<usertoken>buacrj_uyp_iag2s56b7t8opp5edsvt</usertoken>
<actionId>15</actionId>
</qdbapi>
```

URL alternative

```
https://myApp.quickbase.com/db/bmdihpya6?a=API_Webhooks_Copy&actionID=1
```

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
<action>API_Webhooks_Copy</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<actionId>16</actionId>
</qdbapi>
```

Did this help you? Give us a rating:

API_Webhooks_Create

Overview

Use API_Webhooks_Create to set up and configure a webhook.

Request Parameters

PARAMETER	VALUE	REQUIRED ?
Label	A unique name for the webhook	yes
Description	Description of the webhoook	no
Query	Filter criteria to trigger the webhook	no

WorkflowWhen	Condition to trigger the webhook:
	a = add (default)
	d = delete
	m = modify
	These can be combined in any order, for example "mda" (modify, delete, add)
		no

WebhookURL	Endpoint URL for the webhook. This must begin with: *https://* Example: `https://myApp.quickbase.com/db/xxxxxx`	yes

PARAMETER	VALUE	REQUIRED ?
WebhookHeader	Key value pair of the header. A name/value pair that typically describes the format of the message, for example: Content-Type: application/json. If the webhook is going to another Quick Base table, you can provide the API call name (either here or in the URL); for example: Quick Base-Action: API_AddRecord.	No
WebhookHeaderCount	The number of webhook headers. For example: WebhookHeaderCount=1	No
WebhookHeaderKey(n)	For example: WebhookHeaderKey1=TOKEN	No
WebhookHeaderValue(n)	Example: WebhookHeaderValue1=VALUE	No
WebhookMessage	Payload of the webhook (empty by default).	no
WebhookMessageFormat	Format of the payload: XML (default) JSON RAW	no

PARAMETER	VALUE	REQUIRED ?
WebhookHTTPVerb	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response. Specifies the HTTP method for the webhook: POST (default) GET PUT PATCH DELETE	no
tfidsWhich	Set the field IDs criteria to determine if webhook should be triggered. The webhook will fire only if at least one of the indicated fields has changed. Set the value of this parameter to TRUE and for every field ID that should be added to criteria add 'tfids' request parameter. Set the value of this parameter to 'tfidsAny' to clear criteria and fire webhook on any changed field. Example: <tfidsWhich>TRUE</tfidsWhich><tfids>6</tfids><tfids>9</tfids>	No

Response values

PROPERTY TAG	DESCRIPTION
action	Webhooks API name (for example, API_Webhooks_Create)
errcode	numeric error code
errtext	error text, for example: "invalid input"

Sample XML Request

```
POST https://*target_domain*/db/*target_dbid*?
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_Webhooks_Create
<qdbapi>
<usertoken>buacrj_uyp_iag2s56b7t8opp5edsvt</usertoken>
<label>Label</label>
<WebhookURL>https://target_domain/db/target_dbid?</WebhookURL>
</qdbapi>
```

URL alternative

```
https://myApp.quickbase.com/db/bmdihpya6?a=API_Webhooks_Create&Label=New
```

Sample response

Success

```
<qdbapi>
<action>API_Webhooks_Create</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<changed>true</changed>
<success>true</success>
</qdbapi>
```

Failure

```
<qdbapi>
<action>API_Webhooks_Create</action>
<errcode>2</errcode>
<errtext>Invalid input</errtext>
<errdetail></errdetail>
</qdbapi>
```

Did this help you? Give us a rating:

API_Webhooks_Delete

Overview

Use API_ActionWorkFlows_Create to create a webhook.

Request Parameters

PARAMETER	VALUE	REQUIRED?
actionIDList	Comma-separated IDs of the webhooks	yes

Response values

PROPERTY TAG	DESCRIPTION
numChanged	Denotes how many webhooks were deleted.

Sample XML Request

```
 POST https://target_domain/db/target_dbid?
 Content-Type: application/xml
 Content-Length:
 QUICKBASE-ACTION: API_Webhooks_Delete
 <qdbapi>
 <usertoken>buacrj_uyp_iag2s56b7t8opp5edsvt</usertoken>
 <actionIDList>1<actionIDList>
 </qdbapi>
```

URL alternative

```
 https://webhookstest.qblab-
trunk.corp.quickbase.net/db/bmdihpya6?a=API_Webhooks_Delete&actionIDList=1
```

Sample response

```
<?xml version="1.0" ?>
<qdbapi>
<action>API_Webhooks_Delete</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<actionId>1</actionId>
</qdbapi>
```

Did this help you? Give us a rating:

API_Webhooks_Deactivate

Overview

Use API_Webhooks_Deactivate to disable a webhook. To reactivate a webhook, use API_Webhook_Activate.

Request Parameters

PARAMETER	VALUE	REQUIRED?
actionIDList	Comma-separated IDs of the webhook	yes

Response values

ELEMENT NAME	VALUE
action	The originating request, for example, API_Webhooks_Deactivate
errcode	Identifies the error code, if any. (See the Error Codes appendix for a list of possible error codes.) 0 indicates that no error was encountered.
errtext	Text that explains the error code. "No error" indicates that no error was encountered.
numChanged	Denotes the number of webhooks that were deactivated.
success	true or false

Sample XML Request

```
POST https://target_domain/db/target_dbid?
Content-Type: application/xml
Content-Length:
QUICKBASE-ACTION: API_Webhooks_Deactivate
<qdbapi>
<usertoken>buacrj_uyp_iag2s56b7t8opp5edsvt</usertoken>
<actionIDList>15</actionIDList>
</qdbapi>
```

URL alternative

```
https://myapp.quickbase.com/db/bmdihpya6?a=API_Webhooks_Deactivate&actionIDList=
1,2,3
```

Sample response

```
<qdbapi>
<action>API_Webhooks_Deactivate</action>
```

```
<errcode>0</errcode>
<errtext>No error</errtext>
<numChanged>1,2,3</numChanged>
<success>true</success>
</qdbapi>
```

Did this help you? Give us a rating:

API_Webhooks_Edit

- Overview
- Request parameters
- Response values
- Sample XML Request
- URL alternative
- Sample response

Overview

Use API_Webhooks_Edit to edit webhooks.

Request Parameters

PARAMETER	VALUE	REQUIRED ?
actionID	The ID of the webhook action	yes
Label	A unique name for the webhook	yes
Description	Description of the webhoook	no
Query	Filter criteria to trigger the webhook	no

WorkflowWhen	Condition to trigger the webhook:
	a = add (default)
	d = delete
	m = modify
	These can be combined in any order, for example "mda" (modify, delete, add)
		no

PARAMETER	VALUE	REQUIRED ?
WebhookURL	Endpoint URL for the webhook. This must begin with: *https://* Example: `https://myApp.quickbase.com/db/xxxxxx`	yes
WebhookHeader	Key value pair of the header. A name/value pair that typically describes the format of the message, for example: Content-Type: application/json. If the webhook is going to another Quick Base table, you can provide the API call name (either here or in the URL); for example: Quick Base-Action: API_AddRecord.	No
WebhookHeaderCount	The number of webhook headers. For example: WebhookHeaderCount=1	No
WebhookHeaderKey(n)	For example: WebhookHeaderKey1=TOKEN	No
WebhookHeaderValue(n)	Example: WebhookHeaderValue1=VALUE	No
WebhookMessage	Payload of the webhook (empty by default).	No
WebhookMessageFormat	Format of the payload: XML (default) JSON RAW	No

PARAMETER	VALUE	REQUIRED ?
WebhookHTTPVerb	A string value that you want returned. It will not be handled by Quick Base but it will be returned in the response. Specifies the HTTP method for the webhook: POST (default) GET PUT PATCH DELETE	No
tfidsWhich	Set the field IDs criteria to determine if webhook should be triggered. The webhook will fire only if at least one of the indicated fields has changed. Set the value of this parameter to TRUE and for every field ID that should be added to criteria add 'tfids' request parameter. Set the value of this parameter to 'tfidsAny' to clear criteria and fire webhook on any changed field. Example: <tfidsWhich>TRUE</tfidsWhich><tfids>6</tfids><tfids>9</tfids>	No

Response values

PROPERTY TAG	DESCRIPTION
action	Webhooks API name (for example, API_Webhooks_Edit)
errcode	numeric error code
errtext	error text, for example: "invalid input"

Sample XML Request

```
 POST https://target_domain/db/target_dbid?
Content-Type: application/xml Content-Length:
 QUICKBASE-ACTION: API_Webhooks_Edit
 <qdbapi>
 <usertoken>buacrj_uyp_iag2s56b7t8opp5edsvt</usertoken>
 <actionId>15</actionId>
 <label>EditedWebhook</label>
 <WebhookURL>https://myApp.quickbase.com/bin/49b9b88b-5ac9-43ba-93a0-
096f99816afb</WebhookURL>
 <WebhookMessage>Webhook edited version 2</WebhookMessage>
 <WebhookMessageFormat>JSON</WebhookMessageFormat>
 </qdbapi>
```

URL alternative

```
 https://myApp.quickbase.com/db/bmdihpya6?a=API_Webhooks_Edit&actionId=11
```

Sample response

```
 <?xml version="1.0" ?>
<qdbapi>
<action>API_Webhooks_Edit</action>
<errcode>0</errcode>
<errtext>No error</errtext>
<changed>true</changed>
<success>true</success>
</qdbapi>
```

Did this help you? Give us a rating:

Using API endpoints for SCIM provisioning

If you need to provide a custom solution or integration for provisioning users or groups, Quick Base provides SCIM-compliant API endpoints that support creating, reading, updating and deleting (CRUD) users or groups via GET, POST, PUT, PATCH, and DELETE methods.

The following topics provide details on using the SCIM API:

- Setting up a user and user token for SCIM provisioning
- SCIM API endpoint details and syntax for users
- SCIM API endpoint details and syntax for groups
- Restrictions using the SCIM API
- Troubleshooting IdM and SCIM provisioning

Did this help you? Give us a rating:

Setting up a user and user token for SCIM provisioning

To provision users via the SCIM API endpoints, you need to use a Quick Base user with realm admin and account admin privileges. The user can either be an existing, individual user, or a shared system user. See below for guidance on setting up a system user.

The user for provisioning also requires a user token. For steps on creating a user token, read About User Tokens. Once the user token is created, you include the user token in an `Authorization` header with a type of `Bearer`.

Setting up a system user

You can use a shared system user to help ensure that provisioning occurs without interruption.

Note: A system user isn't included counted as a paid Quick Base user unless the user is also used to share apps, like a normal Quick Base user.

If you are not using SAML authentication for your realm, create a system user in Quick Base with realm admin and account admin privileges. Contact Quick Base Customer Care if you need help assigning realm admin privileges to this user.

If you are using SAML authentication for your realm, system user creation varies depending on how your realm is configured:

- If your realm is set to require SAML authentication for all users, create a system user in your SAML directory, and then use your normal process to add the user to Quick Base with realm admin and account admin privileges.
- If your realm is not set to require SAML for all users, you can either use the SAML process or create the user directly in Quick Base.

Did this help you? Give us a rating:

SCIM API endpoint details and syntax for users

The Quick Base API endpoints support the SCIM 2.0 protocol and allows for username creates, reads, updates, and deletes (CRUD) via the following HTTP methods:

- GET
- POST
- PUT
- DELETE

A bearer token is used for authentication using a Quick Base user token. Read Setting up a user and user token for SCIM provisioning for details.

Quick Base supports the following via CRUD:

- Get users in a realm
- Create users/Provision users
- Search for users in app or realm
- Update/edit user information
- Deny/restore users
- Delete users

Note: The Patch endpoint is not supported.

Required user fields are:

- userName
- name
- emails
- active
- externalID

The following sections provide syntax examples.

Get users in a realm

Note: The username eq filter query is supported.

- Get all users: `GET { SCIMBaseURL}/users`

- Get users with pagination: `GET { SCIMBaseURL}/users?count=1&amp;startIndex=1`

- Get users with userId: `GET { SCIMBaseURL}/users/{userId}`

- Get users with userName filter: `GET { SCIMBaseURL}/users?filter=userName eq "{userName}"`

Create users/Provision users

Note: New users are created as approved, registered and verified. Once provisioned, the new users need to sign in to Quick Base via SSO.

```
POST { SCIMBaseURL}/users
  {
     "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
     "userName": “Jane.doe",
     "name": {
         "givenName": "Jane",
         "familyName": "Doe"
     },
     "emails": [{
         "primary": true,
         "value": " jane.doe@example.com"
     }],
     "externalId": "00uv931EiyRsnwOGa0g3",
     "active": true
 }
```

Update/edit user information

```
PUT { SCIMBaseURL}/users/{userId}
  {
     "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
     "userName": “ jane.smith",
     "name": {
         "givenName": "Jane",
         "familyName": “Smith"
     },
     "emails": [{
         "primary": true,
         "value": " jane.smith@example.com"
     }],
     "externalId": "00uv931EiyRsnwOGa0g3",
     "active": true
 }
```

Deny users

```
PUT { SCIMBaseURL}/users/{userId}
  {
     "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
     "userName": “ jane.smith",
     "name": {
         "givenName": "Jane",
         "familyName": “Smith"
     },
     "emails": [{
         "primary": true,
         "value": " jane.smith@example.com"
     }],
     "externalId": "00uv931EiyRsnwOGa0g3",
      "active": false
 }
```

Approve users

```
PUT { SCIMBaseURL}/users/{userId}
  {
     "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
     "userName": " jane.smith",
     "name": {
         "givenName": "Jane",
         "familyName": "Smith"
     },
     "emails": [{
         "primary": true,
         "value": " jane.smith@example.com"
     }],
     "externalId": "00uv931EiyRsnwOGa0g3",
      "active": true
 }
```

Delete users

```
DELETE { SCIMBaseURL}/users/{userId}

 { SCIMBaseURL} = {BaseURL}/ governance/scim/v2/
```

Did this help you? Give us a rating:

SCIM API endpoint details and syntax for groups

The Quick Base API endpoints support the SCIM 2.0 protocol and allow you to create, read, update, and delete (CRUD) groups using the following HTTP methods:

- GET - get groups in a realm/search for groups in an app or realm
- POST - create groups/provision groups
- PUT - update/edit group information
- DELETE - delete groups
- PATCH - patch group information

A bearer token is used for authentication using a Quick Base user token. For more details, select Setting up a user and user token for SCIM provisioning for details.

The groupId used in the following code examples is the group ID created by Quick Base, not the IdM provisioned group ID.

Get groups in a realm

- Get all groups: `GET { SCIMBaseURL}/groups`
- Get groups with pagination: `GET { SCIMBaseURL}/groups?count=1&amp;startIndex=1`
- Get groups with groupId: `GET { SCIMBaseURL}/groups/{groupId}`

Create groups or provision groups

To create or provision a new group, use the following request format:

```
POST { SCIMBaseURL}/groups
  {
     "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
     "displayName": “scimGroup-Test1",
     "members": {
         "value": "{userID}",
                 "$ref": "{Users/userID}",
                 "display": "Jane Doe",
     }],
  }
```

Update or patch groups

To update an existing group, use the following request format:

```
PUT { SCIMBaseURL}/groups/{groupId}
  {
     "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
     "displayName": “scimGroup-Test1",
     "members": {
         "value": "{userID}",
                 "$ref": "{Users/userID}",
                 "display": "Jane Doe",
     }],
  }
```

To patch an existing group by adding members, use the following request format:

Only the members you specified in the request are added to the group memberships.

```
PATCH { SCIMBaseURL}/groups/{groupId}
  {"schemas":
        ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
       "Operations":[
         {
          "op":"add",
          "path": "members",
          "value":[
           {
             "display": "Babs Jensen",
             "$ref":"{Users/userID}",
             "value": "{userID}"
           }
          ]
         }
        ]
       }
```

To patch an existing group by replacing all members, use the following request format:

```
PATCH { SCIMBaseURL}/groups/{groupId}
  {"schemas":
        ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
       "Operations":[
         {
          "op":"replace",
          "path": "members",
          "value":[
           {
             "display": "Babs Jensen",
             "$ref":"{Users/userID}",
             "value": "{userID}"
           },
           {
             "display": "James Smith",
             "$ref":"{Users/userID}",
             "value": "{userID}"
           }
          ]
         }
        ]
       }
```

To remove a member using the "eq" filter, use the following request format:

When you remove a member using the "eq" filter, the members listed in the path with the filter are removed from this group. No other memberships of the group are modified.

```
PATCH { SCIMBaseURL}/groups/{groupId}
  { "schemas":
      ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
    "Operations":[
      {
        "op":"remove",
        "path":
            "members[value eq\"{userID}\"]"
      },
    ]
  }
```

To remove a member without the "eq" filter, use the following request format:

When you remove a member without the "eq" filter, the members listed under "value" are removed from this group. No other memberships of the group are modified.

```
PATCH { SCIMBaseURL}/groups/{groupId}
  { "schemas"
      ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
    "Operations":[
  {
        "op":remove",
        "path": "members",
        "value":[{
               "display": "James Smith",
               "$ref":"{Users/userID}",
               "value": "{userID}"
           },
           {
             "display": "Babs Smith",
             "$ref":"{Users/userID}",
             "value": "{userID}"
           }
          ]
         }
    ]
  }
```

Delete groups

To delete a group, use the following request format:

```
DELETE { SCIMBaseURL}/groups/{groupId}
```

The base URL for SCIM above is: `{ SCIMBaseURL} = {BaseURL}/ governance/scim/v2/`

Did this help you? Give us a rating:

Restrictions using the SCIM API

This topic lists and explains restrictions on provisioning via the Quick Base SCIM API.

Allowable usernames

Usernames cannot have spaces or be longer than 255 characters. Special characters are allowed.

Preventing conflicts with email addresses as usernames

If an email address is specified as a username, Quick Base verifies updates to usernames to prevent conflicts with existing users.

For example, in this scenario, a new user can be provisioned without conflict:

USER	USERNAME	EMAIL ADDRESS
User 1 (existing user)	user@example.com	a@example.com
User 2 (new provisioned user)	user2	a@example.com

But a conflict is caused if an email update occurs as follows, because **user@example.com** already exists as a username.

USER	USERNAME	EMAIL ADDRESS
User 1	**user@example.com**	a@example.com

USER	USERNAME	EMAIL ADDRESS
User 2	user2	**user@example.com**

Sign-in must use SSO

Sign-in to Quick Base must use SSO, because of the method by which usernames provisioned via the SCIM API are stored.

Deletions

Deletions via the SCIM API are mapped as denied users in Quick Base.

Updates to basic information limited to in-company

Updates to basic information (first name, last name, email address) are limited to those within the same company. Updates to basic information for usernames outside the company are not supported.

Link only through primary emails

Quick Base supports linking via the value set for primary emails only.

App sharing

To share Quick Base apps, users must share the app with the email address of the SCIM user, or with the Quick Base username (if the user had already existed in Quick Base before provisioning).

Provisioned existing users

If you use the SCIM API to provision existing users, and the user already has a Quick Base username set, the Quick Base username will be cleared.

Deactivated users

You cannot use the SCIM API to restore any Quick Base users that have a status of deactivated.

Rate limitation

SCIM API usage rate may be limited. If your requests are being limited, an HTTP: 429 error will be returned. Most IdM tools recognize rate limit status and automatically retry.

Restrictions on changing SCIM users from Quick Base

The following activities cannot be completed using the Quick Base Admin Console. You must use an IdM provider or the SCIM API:

- Deny users
- Update email, username, firstname, lastname

Group mapping management restrictions from Quick Base

- You cannot appoint new managers, or promote users to managers.
- You must manage the following items through your IdM application: add or remove SCIM members, and delete or rename SCIM groups.
- You cannot link existing groups in Quick Base to SCIM groups in your IdM application. All SCIM groups are new groups.
- You cannot rename or create a non-SCIM group using an existing SCIM's group name. Each SCIM group name is unique within a realm. For example, you cannot create a non-SCIM group using "QB_PD" because a SCIM group with the same name exists within the same realm:

GROUP TYPE	GROUP NAME	CREATION STATUS
SCIM group	QB_PD	existing group
Non-SCIM group	QB_PD	name conflict

- With subscriptions and email notifications, if an existing user has the same username as a SCIM group in the same realm, then any subscriptions or email notifications created are sent to the user. For example, within the same realm, if there is a user with the username "sales" and a new SCIM group is created with the name "sales", then any subscription or email notification is sent to the user.
- You cannot create nested SCIM groups.

Did this help you? Give us a rating:

Troubleshooting IdM provisioning

ERROR	SOLUTION
Multiple users with the same email address. Message: "Provisioning failed: Multiple users in the realm match the given user's email address."	This message appears when your realm has multiple users that share the same email address. Via the Quick Base Admin Console, edit the correct user to add the SCIM external IdM username, then provision the user again via your IdM or SCIM API solution.

ERROR	SOLUTION
Users exist elsewhere in Quick Base with the same email address. Message: “Provisioning failed: Multiple users in Quick Base match the given user's email address, but none exist in the realm.”	This message appears when there are no users currently in your realm that have this email address, but users exist elsewhere in Quick Base that have the same email address. You need to add the user to your realm and re-provision the user. Contact Quick Base Customer Care if you need help.
The updated user attribute (First name, Last name, email) is not reflected on the Quick Base Manage All Users page.	Quick Base doesn’t allow you to edit users that are not in your company domain. Make sure the user is a user whose email domain belongs to the company or replace the API token used to configure the account with a realm admin who is an in-company user.

Did this help you? Give us a rating:

Appendix A: Error Codes

CODE	MESSAGE	POSSIBLE CAUSE
0	No error	No error
1	Unknown error	
2	Invalid input	You did not specify a name for the new application. To grant Group Record permissions, you must specify the Target Group name. This option is available to corporate workgroup users only. You cannot place more than .5 MB in this field. You do not have permission to access this table. You did not specify the field name. The field identifier (fid) does not refer to an existing field. Please provide a valid field identifier.
3	Insufficient permissions	You do not have permission to perform the current operation. You do not have permission to clone (copy) an application. At a minimum, you must have View permission to copy an application. You do not have permission to query this table. At a minimum, you must have View permission to perform this operation. You do not have sufficient permission to delete the records you specified from this table.

CODE	MESSAGE	POSSIBLE CAUSE
4	Bad ticket	No user account was found that matches the e-mail address (or user name) and password you entered. Your ticket has expired or is invalid. By default tickets expire after 12 hours of use. One-time use tickets expire after one use or after 5 minutes. Please try signing in again. To perform the current operation, you must supply one of the following: A user name and password A ticket To get a list of tables to which you have been granted access, you must supply one of the following: A user name and password A ticket
5	Unimplemented operation	
6	Syntax error	Your Quick Base API call does not have the correct syntax.
7	API not allowed on this application table	
8	SSL required for this application table	
9	Invalid choice	You cannot set the specified field to "(choice)".
10	Invalid field type	
11	Could not parse XML input	Your XML request contains invalid XML.

CODE	MESSAGE	POSSIBLE CAUSE
12	Invalid source DBID	
13	Invalid account ID	
14	Missing DBID or DBID of wrong type	
15	Invalid hostname	
19	Unauthorized IP address	Realm IP filtering is in place, and the user is not logged in from an authorized location.
20	Unknown username/password	No user account was found that matches the e-mail address (or user name) and password you entered.
21	Unknown user	No user account was found that matches the e-mail address or user name you entered. No group was found that matches the group name you entered.
22	Sign-in required	
23	Feature not supported	
24	Invalid application token	
25	Duplicate application token	

CODE	MESSAGE	POSSIBLE CAUSE
26	Max count	
27	Registration required	The user specified is someone who has been invited to join an application, but who has never used Quick Base before.
28	Managed by LDAP	
29	User on Deny list	
30	No such record	There is no record in the table that matches the specified record identifier (rid).
31	No such field	There is no field in this table that matches the specified field identifier (fid).
32	The application does not exist or was deleted	
33	No such query	There is no query in this table that matches the specified query identifier (qid). There is no query in this table that matches the specified query name (qname).
34	You cannot change the value of this field	
35	No data returned	

CODE	MESSAGE	POSSIBLE CAUSE
36	Cloning error	
37	No such report	
38	Periodic report contains a restricted field	
50	Missing required field	The default value property for the specified field is blank. Please supply a non-blank value for this field. You have either provided a blank value for one or more required fields, or you haven't supplied any value for one or more required fields whose default property is blank.
51	Attempting to add a non-unique value to a field marked "unique"	
52	Duplicate field	
53	The following required fields are missing from your import data: <field_1> [[<field_2>]...]	
54	Cached list of records not found	
60	Update conflict detected	

CODE	MESSAGE	POSSIBLE CAUSE
61	Schema is locked	You cannot make development changes to an application because it is locked. Quick Base locks an application when you create a sandbox copy of the application.
70	Account size limit exceeded	
71	Database size limit exceeded	
73	Your account has been suspended	
74	You are not allowed to create applications	You do not have Create permissions on the billing account.
75	View too large	
76	Too many criteria	The limit on the number of criteria is currently set to 100 and you have exceeded this limit.
77	API request limit exceeded	This can be returned if an API is called too frequently. The same API can't be called again before the time specified in the error message.
78	Data limit exceeded	
80	Overflow	
81	Item not found	

CODE	MESSAGE	POSSIBLE CAUSE
82	Operation took too long	
83	Access denied	
84	Database error	
85	Schema update error	
87	Invalid group	No group matched the group ID you provided.
100	Technical Difficulties -- try again later	
101	Quick Base is temporarily unavailable due to technical difficulties	
102	Invalid request - we cannot understand the URL you specified	
103	The Quick Base URL you specified contained an invalid srvr parameter	
104	Your Quick Base app is experiencing unusually heavy traffic. Please wait a few minutes and re-try this command.	

CODE	MESSAGE	POSSIBLE CAUSE
105	Quick Base is experiencing technical difficulties	
110	Invalid role	The specified role does not exist in the Quick Base.
111	User exists	The specified user already exists.
112	No user in role	If you try to change a user’s role and no user is in the role you specify as the user’s current role, you’ll get this error.
113	User already in role	The user has already been assigned the role you are trying to assign.
114	Must be admin user	
150	Upgrade plan	
151	Expired plan	
152	App suspended	The application is suspended, possibly due to nonpayment.

Did this help you? Give us a rating:

Appendix B: Managing files

Quick Base applications can support file attachment storage and file attachment access through the API's file upload and download feature. Files are "attached" through a record field that has the type File Attachment.

File attachments are stored in a special database that is separate from the Quick Base application database. That means they won't take up space in the Quick Base application database.

This section describes the typical upload scenario where access to the local file system is available if needed. If your environment cannot make use of the local file system, you may need to use the API_UploadFile call.

Uploading files

Files are uploaded as file attachments to a record during an API_AddRecord or API_EditRecord call. The file field you use for file attachments must be of type File Attachment.

The file must be base64-encoded. You should not use MIME style encoding with newline characters at a maximum line length of 76. Instead, Quick Base requires you omit these newlines. (If you don't omit the newlines, the stored file in Quick Base won't be usable.)

Because of the typical file sizes, you won't be able to upload files using a URL. You must POST the encoded file as XML.

Downloading files

You can download a file in two ways:

- If you know the database ID, the record ID, the Field ID of the file field you are attaching to, and the version ID, you can use issue a GET request using this URL format:
  `https://`*target_domain*`.quickbase.com/up/`**DBID**`/a/r`**RID**`/e`**FID**`/v`**VID**

  where:

  - *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.
  - **DBID** is your table dbid (not the application dbid)
  - **RID** is the record ID of the record with the file attachment
  - **FID** is the field ID of the file attachment field
  - **VID** is the version ID of the file. Specifying 0 (zero) for the VID always gets you the most recent version.

  For example: `https://quickbase.com/up/bdb5rjd6h/a/r13/e8/v0`

- If you know the URL of the file attachment, you can issue a GET request that contains the URL of the file attachment and the table dbid. (To get the URL, use API_DoQuery if and set its **fmt** parameter to "structured." The URL is returned in the `<url></url>` tags.)
  Here's a typical URL that you would use in this second way of doing a GET:

  `https://`*target_domain*`.quickbase.com/up/bdb5rjd6g/g/rz/ey/va/Model_T.jpg`

Note: If you use file attachments, you should understand how Quick Base versions files. See the Quick Base online help for more information.

Did this help you? Give us a rating:

Appendix C: Building and Using Queries

Out of all of the HTTP API calls, only two calls perform the critical function of returning the data you need in the display order (view) that you want. These two calls are API_DoQuery and API_GenResultsTable: both of which are covered in Quick Base API Call Reference section.

Both API_DoQuery and API_GenResultsTable make use of the same query mechanism to return data. They differ mainly in the overall format of the return:

- API_DoQuery returns XML that you can parse and process as you wish.
- API_GenResultsTable can be embedded in a web page to automatically show the filtered records in the page as an HTML table when the page is opened in a browser.

See the following topic for more information:

- Components of a query

Did this help you? Give us a rating:

Components of a query

A query typically includes the following parameters:

- Query parameter
- Column list
- Sort list
- Options
- Format

The Query parameter

The query parameter determines which records are returned. You can build a new query string inside the <query></query> tag or you can choose to reference a saved query string using the *<qid>* or *<qname>* tags.

Building the query string

When you build a query, you must build at least one query string. A query string is composed of:

- a field ID (fid)
- an uppercase comparison operator (see the table below for a list of available operators).
- the value to be compared against

You should separate each of these query string components using a period and enclose the entire query string in curly braces, as shown below:

```
{'fid'.operator.'matching_value'}
```

Example:

```
<query>{'5'.CT.'Ragnar Lodbrok'}</query>
```

This example specifies that Quick Base should return all records where the “fid 5” field contains the value “Ragnar Lodbrok.”

Grouping multiple query strings

You can group several of query strings together, separating the query strings with the AND or OR operators, as shown here:

Example:

```
<query>{'5'.CT.'Ragnar Lodbrok'}AND{'7'.CT.'Acquisitions'}</query>
```

The next example illustrates how to group query strings together when using a URL:

Example:

```
&query={'5'.CT.'Ragnar Lodbrok'}AND{'7'.CT.'Acquisitions'}
```

Note: You can obtain the fids for a table by calling API_GetSchema, or from any call to API_DoQuery that sets the **fmt** parameter to “structured.”

Query string comparison operators

Note: Query string comparison operators must be in uppercase.

COMPARISON OPERATOR	DESCRIPTION
CT	Contains (Do not use this operator with list-user fields; instead, use HAS.)
XCT	Does not contain (Do not use this operator with list-user fields; instead, use XHAS.)

COMPARISON OPERATOR	DESCRIPTION
HAS	Contains a specific set of users. Used with list-user fields only. For each user you are trying to find, you must enter the user's ID, user name, or email address. You can also enter placeholder names. Be sure to surround placeholder names with double quotes. The query parameter must be surrounded by single quotes. Separate users in the list using a semi-colon. For example: `<query>{'6'.HAS.'-8675309; -9873297'}</query>`
XHAS	Does not contain a specific set of users. Used with list-user fields only. The query parameter must contain the user's ID, email address, or user name. You can also specify a placeholder name. Placeholder names must be enclosed in double quotes. The entire query parameter must be surrounded by single quotes. Separate users in the list using semi-colons. Note that a matching record must contain all users you specify. This query: `<query>{'6".XHAS. '-8675309; -9873297'}</query>` ...specifies that you want to see records that do not contain BOTH of these users. Therefore, the query will return records that contain either one or neither, but not both, of these users.
EX	Is
TV	True Value (compares against the underlying foreign key or record ID stored in relationship fields). Also used for queries on User fields.
XTV	Not True Value (compares against the underlying foreign key or record ID stored in relationship fields). Also used for queries on User fields.

COMPARISON OPERATOR	DESCRIPTION
XEX	Is not
SW	Starts with
XSW	Does not start with
BF	Is before
OBF	Is on or before a specific date
AF	Is after a specific date
OAF	Is on or after a specific date
IR	Is in range. Use this operator with date fields, to determine whether a particular date falls within particular date range relative to the current date. Learn more about relative date ranges.
XIR	Is not in range. Use this operator with date fields, to determine whether a particular date falls within a particular date range relative to the current date. Learn more about relative date ranges.
LT	Is less than
LTE	Is less than or equal to
GT	Is greater than

COMPARISON OPERATOR	DESCRIPTION
GTE	Is greater than or equal to

Sample queries

To query a particular field and exclude a string:

```
{'7'.XCT.'Fred'}
```

To query any field for a date:

```
{'0'.OAF.'3/31/01'}
```

To combine multiple query strings:

```
{'9'.SW.'Wilma'}OR{'10'.XSW.'Dino'}
```

To query for records owned by the current user:

```
{'4'.TV.'_curuser_'}
```

To query for records modified by the current user:

```
{'5'.TV.'_curuser_'}
```

Handling Special Characters for a Query Field Values

If you are searching for a value that includes special characters, be sure to enclose the matching value in quotes. For example, if you are searching for this value:

"Joe B. Briggs"

...you should be sure to enclose the entire string in double quotes, as follows:

""Joe B. Briggs""

Querying on Dates and Times

You can query on dates (yyyy-MM-dd) or on times in milliseconds, but you cannot query on date-time values in the standard "yyyy-MM-dd hh:mm:ss" format.

Special Matching Values in a Query String

Certain field types use special values with a given comparison operator. For example, you may want to query on a Checkbox field. You may want all records where a Checkbox is either checked or not. In cases like this, you can use special matching values in your query string to return records with the value you want.

FIELD TYPE	DESCRIPTION
Checkbox	1 or 0
Date and Date/Time	today or <num> days ago or -<num> days ago or mm-dd-yyyy
User	_curuser_

Referencing a Common Report

In the Quick Base UI, a user can customize a report and save it as a common report, which will then be available as a saved query. Each saved query has an ID (qid) and a user-supplied name (qname). Creating a saved query is an easy and quick way to construct a query, compared to building the query string yourself.

To use a saved query, you refer to it by its qid (e.g., `<qid>3</qid>`) or by the name you gave it when you saved it (e.g., `<qname>My Custom View</qname>`. For the qname value, do not replace blanks spaces with underscore.

You can use API_DoQuery to get the qid and qname of common reports. If you use the **fmt** parameter, Quick Base returns a list of all saved queries inside a <queries> aggregate, including their qid and qnames. (You can also use API_GetSchema to get the qid or qname of a saved query).

Quick Base tables have two default saved queries: List All, and List Changes.

Column List (clist) Parameter

The clist specifies which columns will be included in each returned record and how they are ordered in the returned record aggregate.

XML Example:

```
<clist>4.7.9.11</clist>
```

URL Example:

```
&clist=4.7.9.1
```

...where each number is the fid of the field (column) you want returned, with a period separating the fids. The columns will be listed in the returned record in the same order as they appear in the clist. If you don't specify a clist, Quick Base returns your table's default columns and order.

If no column list parameter is supplied, the default report layout and field order is used.

To change the default field order, see the Quick Base online help:

- Changing the default field order in a table.
- Changing the default field order on a form.

How to Return ALL of the columns

Specify the value *a* in the **clist** param to specify that all columns should be returned.

Sort List (slist) Parameter

The sort list parameter determines the order in which the returned records are displayed as follows:

```
<slist>fid</slist>
```

The following **slist** parameter sorts all records by the field whose fid is 7.

XML Example:

```
<slist>7</slist>
```

You can specify a secondary sort by including another fid in the string (separating each fid with a period)

XML Example:

```
<slist>7.12</slist>
```

URL Example:

```
&slist=7.12
```

If you don't specify the sort list, the default sort specified on the Report Layout page determines the order in which records are displayed.

Options Parameter

You use the **options** parameter to further control the return of records. You can use any or all of the options, separating the options with a period.

- `num-n` --specifies a maximum of n records in the return
- `onlynew` --returns only those records marked with new or updated flags
- `skp-n` --skips the first n records returned by the query
- `sortorder-A` --specifies an ascending order
- `sortorder-D` --specifies a descending sorts.

XML Example:

```
<options>num-15.sortorder-A.skp-15</options>
```

URL Example:

```
&options=num-15.sortorder-A.skp-15
```

This tells Quick Base to skip the first 15 records the query would normally return and give you the next 15 records. It also specifies that all sorts should be in ascending order.

If you don't supply the **options** parameter, the records are sorted in ascending order and the number of records returned will be the same number as the Quick Base is set to return.

About the Sort Order Option

You can apply a different sort order to different returned columns in the clist by supplying several period-delimited **sortorder** parameters:

```
<options>sortorder-A.sortorder-D.sortorder-D.sortorder-A</options>
```

This example sorts the first column in the clist in ascending order, the second and third columns in descending order, and the fourth column in ascending order.

The **onlynew** parameter produces a different result for each user that is logged into Quick Base: it won't work with the Quick Base anonymous user.

Format Parameter

Use the **fmt** parameter to specify structured returns. If you don't specify this parameter, query returns are **not** structured; Quick Base returns the table name, any DBVars, and the record data with the fields you specified, all with the field names, like this:

```
<record>
   <business_phone_number>(123) 103-1234 x12345</business_phone_number>
   <email>b@c.com</email>
   <file>car.jpg</file>
   <text>justatest</text>
   <update_id>1206568990479</update_id>
</record>
```

In contrast, if you specify structured returns using the **fmt** parameter, Quick Base returns field IDs (fids) instead of field names. Quick Base will also return URLs of file attachments. (For an alternative way to download files without this URL, see Downloading Files.) Using structured format, Quick Base also returns:

- table dbid
- metadata information
- a list of all saved queries you can use
- field properties and the default values for all the fields you asked for in your query.

When you use structured format, the field data is presented in a more compact form so you can immediately spot the fid and the actual field value.

```
<record>
   <f id="5">112149.bhsv</f>
   <f id="6">(650) 123-1234</f>
   <f id="7">cu@later.com</f>
   <f id="22"> Model_T.jpg
```

```
      <url>https://target_domain.quickbase.com/up/bdb5rjd6g/g/rz/ey/va/Model_T.jpg
      </url>
   </f>
   <update_id>1206568990479</update_id>
</record>
```

In the example, *target_domain* is the domain against which you are invoking this call, for example, *quickbase.com*. Read about this notation.

Did this help you? Give us a rating:

Appendix D: APIs and International Use of Quick Base

There are several Quick Base settings used to support the needs of our customers around the world: currency symbols, their placement relative to a number, number formats (including digit grouping, thousands separator, and decimal separator), and date formats. For more information on the use of these settings, see:

- About Localizing Number Formats and Separators
- About Localizing Currency Symbols
- About Localizing Dates

Numeric and Numeric-Currency fields

API_AddRecord and API_EditRecord

API_AddRecord and API_EditRecord allow entering and editing Numeric field data in a format consistent with the field properties. Number formatting included with data in these API calls is ignored, similar to API_ImportFromCSV.

API_CloneDatabase

API_CloneDatabase preserves the number formats, currency symbols, and currency symbol placements at both the app and field level.

API_GetRecordInfo

API_GetRecordInfo includes the numeric value formatted according to the field's properties (in the **printable** element) and the raw numeric value, formatted as 12345678.00 (in the **value** element). As always, the **printable** element is not returned if it is the same as the **value** element.

API_GetSchema and API_DoQuery

API_GetSchema's and API_DoQuery's **&fmt=structured** parameter return the currency symbol, symbol placement, and number format field properties for Numeric fields.

Quick Base supports the following six number formats, which are combinations of digit grouping (2 or 3), thousands separator (, or .), and decimal separator (, or .). The value representing the current number format is returned in the **numberfmt** element. Values are:

VALUE	DECIMAL SYMBOL AND NUMBER SEPARATOR
0	12345678.00
3	12,345,678.00

VALUE	DECIMAL SYMBOL AND NUMBER SEPARATOR
6	12.345.678,00
7	12345678,00
8	1,23,45,678.00
10	1.23.45.678,00

The number representing the currency symbol placement is returned in the **currency_format** element. Values are:

VALUE	CURRENCY SYMBOL PLACEMENT
0	before the number ($-1.00)
1	between the "-" and the number (-$1.00, default)
2	after the number (-1.00$)

API_DoQuery with filter criteria allows the user to search for numeric values with currency symbols and numeric separators as defined by field properties, as well as for unformatted numeric values (no format except decimal separator as defined by the field property). Numeric values are returned formatted as 12345678.00.

HTML-generating APIs

API_GenAddRecordForm, API_GenResultsTable, and API_GetRecordAsHTML allow the user to enter and display numeric data formatted in a fashion consistent with the field properties.

API_ImportFromCSV

API_ImportFromCSV supports round-trip of numeric data. When exporting a CSV file, Quick Base formats Numeric fields according to their field properties. When importing a CSV file, Quick Base uses the properties of the Numeric fields being imported into and ignores number formatting included in the CSV file.

Note: When importing a CSV file to create a table or app, this behavior means that Quick Base uses the default properties for Numeric fields (account-level when creating an app, app-level when creating a table).

API_SetFieldProperties

API_SetFieldProperties can set all currency symbols and number formats we support (see values above). Currency symbols should be XML-encoded.

Date Fields

API_AddRecord and API_EditRecord

API_AddRecord and API_EditRecord allow entering and editing dates in a format consistent with the app property. For example, if the default date format for the app is YYYY-MM-DD then the user should be able to enter a date value with this formatting using these APIs.

API_CloneDatabase

API_CloneDatabase preserves the date formats at the app level.

API_GetRecordInfo

API_GetRecordInfo includes the date's value formatted according the app property (in the **printable** element) and the raw date value formatted in milliseconds since January 1st 1970 00:00:00, UTC (in the **value** element). As always, the **printable** element is not returned if it is the same as the **value** element.

API_GetSchema and API_DoQuery

API_GetSchema now returns the app’s current date format using the **date_format** element:

```
<date_format>YYYY-MM-DD</date_format>
```

API_DoQuery with filter criteria allows the user to search for formatted date values, so long as those values are formatted consistent with the app property. Dates are returned in milliseconds since January 1st 1970 00:00:00, UTC.

HTML-generating APIs

API_GenAddRecordForm, API_GenResultsTable, and API_GetRecordAsHTML allow the user to enter and edit date values formatted in a fashion consistent with the app property.

API_ImportFromCSV

API_ImportFromCSV supports round-trip of dates. When exporting a CSV file, Quick Base formats Date fields according to the app property. When importing a CSV file, Quick Base formats dates according to the app property.

Note: When importing a CSV file to create an app, this behavior means that Quick Base uses MM-DD-YYYY format by default, unless a different default date format has been specified at the account level.

Did this help you? Give us a rating:

Index