



## Cherwell REST API (CSM 9.5.0)

---

## Legal Notices

© 2023 Cherwell Software, LLC. All Rights Reserved.

Cherwell, the Cherwell logo, and mApp are trademarks owned by Cherwell Software, LLC and are registered and/or used in the United States and other countries. ITIL® is a registered trademark of AXELOS Limited. All other product or company names referenced herein are used for identification purposes only and are or may be trademarks or registered trademarks of their respective owners.

Some or all parts of the mApp product are covered by one or more claims of U.S. Patent No. 9, 612, 825.

The information contained in this documentation is proprietary and confidential. Your use of this information and Cherwell Software products is subject to the terms and conditions of the applicable End-User License Agreement and/or Nondisclosure Agreement and the proprietary and restricted rights notices included therein.

You may print, copy, and use the information contained in this documentation for the internal needs of your user base only. Unless otherwise agreed to by Cherwell and you in writing, you may not otherwise distribute this documentation or the information contained here outside of your organization without obtaining Cherwell's prior written consent for each such distribution.

The Cherwell Software product suite includes:

- Cherwell Service Management
- Cherwell Asset Management

[Contact Cherwell Software](#)

# Contents

- Cherwell REST APIs. . . . . 5**
- **About Cherwell REST APIs. . . . . 6**
- **About the Cherwell Canonical REST API. . . . . 10**
- Canonical REST API Mapping Wizard. . . . . 11
- Run a Health Check for Canonical Compliance. . . . . 12
- Interpreting the Results of a Canonical Compliance Health Check. . . . . 13
- Canonical REST API Usage and Examples. . . . . 14
- **Opening the REST API Discovery Tool. . . . . 15**
- Practice Using the REST API Discovery Tool. . . . . 16
- Practice Exercise: Get Search Items. . . . . 17
- Practice Exercise: Get Business Object Schema. . . . . 18
- Practice Exercise: Get Quick Search Configuration for Business Objects. . . . . 19
- Practice Exercise: Save User. . . . . 20
- **Securing the REST API. . . . . 23**
- OAuth2 Protocol for the REST API. . . . . 24
- Obtaining API Client IDs. . . . . 26
- OAuth2 Authentication Modes. . . . . 27
- Authenticating Using the Internal Mode. . . . . 28
- Authenticating Using the Client-Based Windows/LDAP Mode. . . . . 30
- Authenticating Using the Windows/LDAP Mode With Entered Credentials. . . . . 32
- Requesting a Refresh Token. . . . . 34
- SAML Protocol for the REST API. . . . . 36
- Configuring the REST API for SAML Authentication. . . . . 40
- Client Example: No Final URI. . . . . 41
- Client Example: With Final URI. . . . . 44
- **Usage and Examples. . . . . 46**
- Business Object Usage and Examples. . . . . 47
- C# Example: Get an Incident. . . . . 48
- PowerShell Example: Get an Incident. . . . . 50
- C# Example: Create an Incident . . . . . 52
- PowerShell Example: Create an Incident. . . . . 56
- C# Example: Upload Attachments. . . . . 62
- PowerShell Example: Upload Attachments. . . . . 64
- C# Example: Delete a Business Object. . . . . 66
- PowerShell Example: Delete A Business Object. . . . . 68
- Using Search Operations. . . . . 70
- C# Example: Get Search Items. . . . . 73
- PowerShell Example: Get Search Items. . . . . 75
- C# Example: Perform an Ad-hoc Search with Filter. . . . . 77

---

◦ PowerShell Example: Perform an Ad-hoc Search with Filter. . . . .	80
◦ C# Example: Perform an Ad-hoc Search with Sorting. . . . .	83
◦ PowerShell Example: Perform an Ad-hoc Search with Sorting. . . . .	86
◦ Security Usage and Examples. . . . .	88
◦ C# Example: Create a User. . . . .	89
◦ PowerShell Example: Create a User. . . . .	96
◦ Operation Versioning. . . . .	102
◦ Errors. . . . .	103
◦ Using Swagger Code Generation. . . . .	104
◦ Operations List. . . . .	106
◦ Business Object Operations. . . . .	107
◦ Canonical Operations. . . . .	116
◦ Core Operations. . . . .	119
◦ Form Operations. . . . .	123
◦ Named Object Operations. . . . .	124
◦ Add Operations for Named Objects. . . . .	126
◦ Queue Operations. . . . .	127
◦ Search Operations. . . . .	129
◦ Security Operations. . . . .	133
◦ Service Operations. . . . .	137
◦ Team Operations. . . . .	138
◦ User Operations. . . . .	141
◦ Parameter Glossary. . . . .	143

# Cherwell REST APIs

Cherwell REST APIs provide programmatic access to many CSM functions via an HTTP-based RESTful API. Operations are available for:

- Finding, creating, and updating Business Objects
- Finding and running Search queries
- Managing Users
- Getting Mobile Forms

# About Cherwell REST APIs

The [REST API Discovery Tool](#) provides comprehensive API documentation with an intuitive user interface that enables you to discover and test operations using your CSM data. Testing is available for all authentication modes, except SAML.

## REST API Revision History

CSM Version	Changes
9.5.0	<ul style="list-style-type: none"><li>• Added the Canonical REST API Mapping Wizard to simplify the process of mapping a canonical Business Object Schema to your custom Business Object Schema.</li><li>• Added operation for retrieving mapped Activity types for Audit, Communication, and Notes.</li><li>• Removed installed documentation. Users are now directed to the version-specific REST API documentation on <a href="https://help.cherwell.com">https://help.cherwell.com</a>.</li></ul>

CSM Version	Changes
9.4.0	<p>Added Named Object Operations to perform basic functions against Business Objects via the Cherwell REST API. Named Object Operations include nine API operations with simplified URL paths to provide user-friendly, programmatic access to basic CSM functions. You may want to use the Named Object Operations if you are new to the Cherwell REST API or you do not have extensive programming experience. Functions that you can perform using the Named Object Operations include:</p> <ul style="list-style-type: none"> <li>• Delete Business Object</li> <li>• Get Business Object</li> <li>• Update Business Object</li> <li>• Create Business Object</li> <li>• Get Business Object Field List</li> <li>• Delete Business Object Comment</li> <li>• Get Business Object Comments</li> <li>• Create Business Object Comment</li> <li>• Search for Business Object</li> </ul> <p>Added the Cherwell Canonical REST API. The Cherwell Canonical REST API enables customers and partners to create reusable integrations between third-party software and customized CSM systems by providing a fixed schema to map out-of-the-box Business Objects to customized Business Objects.</p>
9.3.0	Minor bug fixes.

CSM Version	Changes
9.2.0	<p>Added operations for:</p> <ul style="list-style-type: none"> <li>• Removing Customers from Workgroups.</li> <li>• Saving Workgroup Members and Team Members.</li> <li>• Getting Teams using the Team ID.</li> <li>• Deleting Teams; removing Users from Teams.</li> <li>• Getting Users using the Record ID; getting a list of all Users.</li> <li>• Adding items to Queues; checking items in to Queues.</li> <li>• Removing items from Queues; checking items out of Queues.</li> <li>• Getting Queues using Scope, Scope Owner, and Folder.</li> <li>• Getting Stored Values using Scope, Scope Owner, and Folder.</li> <li>• Creating, updating, and deleting gallery images.</li> <li>• Getting gallery images using Scope, Scope Owner, and Folder.</li> <li>• Setting the culture for the current User.</li> </ul> <p>In addition, added multiple Powershell examples to the documentation. Expanded information about using OAuth2 and SAML protocols with the REST API. Added SAML client examples with and without a final URI query string. Added practice exercises for using the REST API Discovery Tool. Reorganized the Operations List to match Swagger by adding new categories for Team Operations, User Operations, and Queue Operations. Added instructions for re-generating Swagger code with the new categories after upgrading.</p>
9.1.0	<p>Added operations for:</p> <ul style="list-style-type: none"> <li>• Creating and updating Teams and Workgroups.</li> <li>• Adding a batch of users to Teams.</li> <li>• Getting a Business Object record based on its associated scan code.</li> </ul> <p>In addition, added support for HTML content in Business Object fields.</p>
9.0.1	<p>Added multiple C# examples to the documentation.</p>



CSM Version	Changes
8.3.1	Added v2 of most Security operations to change Error property to ErrorMessage. Added v3 of getuserbyloginid to change Error property to ErrorMessage.
8.2.1	Added operations for: <ul style="list-style-type: none"><li>• Executing Quick Searches</li><li>• Running Saved Queries that have Prompts</li><li>• Adding users to Teams; removing users from Teams</li><li>• Getting Mobile Forms</li></ul> Improved Landing Page that includes parameter glossary and Swagger code-gen instructions.
8.1.1	Added v2 of getuserbyloginid. Changed API version to match CSM version.
8.1.0	Support for Windows, LDAP, and SAML authentication. Attachment management and built-in image retrieval operations added.
8.00	Initial Release.

## About the Cherwell Canonical REST API

The Cherwell Canonical REST API enables customers and partners to create reusable integrations between third-party software and customized CSM systems by providing a fixed schema to map out-of-the-box Business Objects to customized Business Objects.

For example, for a system with a custom Business Object called Ticket, you can leverage an integration with the out-of-the-box, or canonical, Business Object Incident by using the Canonical REST API to map from the Incident Schema to the Ticket Schema.

The Canonical REST API is primarily intended for customers and vendors to create reusable integrations between third-party software and customized CSM systems. If you are a Cherwell Administrator and need to integrate with your instance of CSM, we recommend beginning with the Named Object Operations.



**Important:** To ensure standardization, canonical definitions are supplied and managed by Cherwell. You can access canonical definitions in the Canonical Definitions Manager, but we strongly advise against creating, editing, or deleting canonical definitions as you may inadvertently impact your system's integration capabilities. Canonical definitions are delivered through the mApp Exchange.

# Canonical REST API Mapping Wizard

You can prepare your system for third-party integrations with a custom Business Object by mapping a canonical Business Object Schema to your custom Business Object Schema using the Canonical REST API Mapping Wizard. You can then use the Canonical REST API to programmatically access CSM functions for your custom Business Object.

To use the Canonical Mapping Wizard, perform the following steps:

1. In CSM Administrator, navigate to **Tools > Canonical Mapping Wizard**  
The Map Canonical Object Wizard dialog opens.
2. Select a Canonical Definition from the drop-down menu.  
The list of fields in the Canonical Definition appears.
3. Select the Business Object you want to map.  
You can now select fields from the drop-downs in the Business Object Field area.
4. To create the mappings, you can select the Auto Map button
  - a. Auto Map: When you click the **Auto Map** button, the system will attempt to map Canonical fields to Business Object fields based on the field name. The field names are case-sensitive and must exactly match for the auto mapping to work.  
If the Wizard does not find an exact field name match, the drop-down menu for that field will show FIELD NOT MATCHED. You will need to manually select a field.
  - b. Manually Map: Choose a mapping from the drop-down menu.



**Note:** You must map each field in the Canonical Definition before moving to Step 5.

5. Click the **Next** button. If you are using a Canonical Business Object Schema that requires comments, use this screen to set the mapping. You can use the Auto Map function or manually map the fields.



**Tip:** Journal is the out-of-the-box Business Object that is commonly used to contain comments.

6. Click the **Next** button to create a Blueprint for the mapping.  
The dialog box will show a summary of the changes you have made.
7. Repeat the above steps for each Canonical Definition you want to map.
8. Once you have mapped the fields for each Canonical Definition, publish the Blueprint, then [run a Health Check for canonical compliance](#) to ensure your mapping was successful.
9. You can use the Mapping Wizard to overwrite existing mappings at any time. If you use the Auto Mapping feature, the Wizard will attempt to overwrite all existing mappings for that Canonical Definition.

## Run a Health Check for Canonical Compliance

The Health Check for canonical compliance evaluates mappings between canonical Business Object Schemas and custom Business Object Schemas. It returns errors if the mappings are incomplete or incorrect. Use the Health Check to verify that your system is ready to connect to the Canonical REST API.

1. From the CSM Administrator main window, click the **Performance** category.
2. Select **Run Health Check**.
3. Uncheck all boxes except **Check Canonical Compliance**.
4. Click **OK**.

The Health Check Results open in a separate window. Click **Save to File** to save the results to an HTML file. Click **Submit** to send the results to Cherwell Support on request.

# Interpreting the Results of a Canonical Compliance Health Check

Use the results of your canonical compliance Health Check to verify whether you successfully mapped a canonical Business Object Schema to a custom Business Object Schema. The Health Check results indicate Fields that were mapped successfully as well as Fields that encountered errors.

When you run the Health Check for canonical compliance, your results open in a new window. The results show each mapping that is defined between a canonical Business Object Schema and a custom Business Object schema.

Results in black indicate successful mappings and show the mapped Fields for each schema. Results in red indicate errors with one or more Field mappings. Common errors include:

Error Message	Description
Missing one or more field maps: [Field]	The canonical Business Object Schema includes the indicated Field, and it is not mapped to a Field in the custom Business Object Schema.
[Field] => [Field] (does not exist)	The mapped Field does not exist. Verify that the Field exists and ensure the correct spelling and case.
[Field](Type) => [Field](Type) (FieldType Mismatch)	The mapped Fields are not of the same type; for example, you cannot map a logical Field to a number Field.
[Field](Source field required but not mapped)	The custom Business Object Schema includes the indicated, required Field, and it is not mapped to a Field in the canonical Business Object Schema.
[Field](ReclId) not mapped to canonical	The custom Business Object Schema includes a Record ID Field that is not mapped to a Field in the canonical Business Object Schema.

# Canonical REST API Usage and Examples

The Cherwell Canonical REST API supports basic CRUD (create, read, update, and delete) operations on Business Objects and Business Object comments. You can also perform search operations to search for Business Objects. Code samples for the Canonical REST API are available in [C#](#).

The Canonical REST API is primarily intended for customers and vendors to create reusable integrations between third-party software and customized CSM systems. If you are a Cherwell Administrator and need to integrate with your instance of CSM, we recommend beginning with the Named Object Operations.

Before you begin programming, verify that your system is set up to connect to the Canonical REST API. Refer to [Canonical REST API Mapping Wizard](#) for procedural information.

C# code samples for the Canonical REST API are provided on Cherwell's GitHub page in the [CanonicalAPI-Samples repository](#).

# Opening the REST API Discovery Tool

The Cherwell REST API Discovery Tool (Swagger) can be found by appending the following parameters to the CSM site name URL:

/CherwellApi/swagger/ui/index

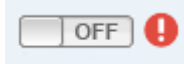
## Logging in to the Discovery Tool

To log in to the Discovery Tool, you need:

- An API client ID. See [Obtaining API Client IDs](#).
- A CSM login ID and password.
- An API Client Secret ID (reserved for Cherwell Software use only).

To log in to the Discovery Tool:

1. Enter the key into the **client\_id** box.
2. Click the Authentication button for a specific operation, and then provide your CSM credentials.



# Practice Using the REST API Discovery Tool

Become familiar with the REST API Discovery Tool (Swagger) before you begin coding so that you can build a program that interacts effectively with CSM. Practice calling operations and interpreting the responses so that you are prepared to write and troubleshoot your program.

No experience with APIs is necessary to perform the practice exercises, though some programming experience will make it easier for you to grasp the concepts. A strong background in CSM is highly recommended. Work closely with your system administrator if your CSM experience is limited.



**Important:** Use a test environment to perform practice exercises.

## Getting the Most From the Practice Exercises

To maximize your understanding, we recommend that you progress through the exercises in order of difficulty, beginning with the simplest:

1. [Practice Exercise: Get Search Items](#)
2. [Practice Exercise: Get Business Object Schema](#)
3. [Practice Exercise: Get Quick Search Configuration for Business Objects](#)
4. [Practice Exercise: Save User](#)

## Practicing on Your Own

In addition to the provided practice exercises, you may practice calling any of the API operations on your own by inputting the required parameters and generating JSON responses. To help interpret response data, consider using an online JSON viewer.



## Practice Exercise: Get Search Items

This is the first in a series of practice exercises to help you become familiar with Cherwell's REST API Discovery Tool. In this exercise, you will call a simple operation with a single, boolean parameter.

1. Log into the REST API Discovery Tool. See [Opening the REST API Discovery Tool](#).
2. Click the **Searches** heading to expand the list of operations.
3. Click **/api/V1/getsearchitems** to expand the operation's details.
4. Under parameters, select **true** for the links parameter.
5. Click **Try it out!**  
If the response data does not appear, ensure that the authentication button has been switched to **On**.
6. **Optional:** Try calling the operation again using **false** for the links parameter to see the difference in response data.

## Practice Exercise: Get Business Object Schema

This is the second in a series of practice exercises to help you become familiar with Cherwell's REST API Discovery Tool. We highly recommend that you complete the following exercises before you begin:

- [Practice Exercise: Get Search Items](#)

In this exercise, you will call the Get Business Object Summary operation to get the Business Object ID for Incident. Then, you will use the Business Object ID to call the Get Business Object Schema operation.

Get the Business Object ID:

1. Log into the REST API Discovery Tool. See [Opening the REST API Discovery Tool](#).
2. Click the **Business Objects** heading to expand the list of operations.
3. Click `/api/V1/getbusinessobjectsummary/busobname/{busobname}` to expand the operation's details.
4. Under parameters, type Incident for the busObName parameter.
5. Click **Try it out!**  
If the response data does not appear, ensure that the authentication button has been switched to **On**.
6. From the response body, copy the value for busObId. This is the Business Object ID that you will use to call the next operation.

Use the Business Object ID to get the Business Object schema:

7. Click `/api/V1/getbusinessobjectschema/busobid/{busobid}` to expand the operation's details.
8. Under parameters, paste the Business Object ID in the busObId Value field.
9. Select **false** for the includeRelationships parameter so that the response data will not include information about related Business Objects.
10. Click **Try it out!**
11. **Optional:** Try calling the operation again using a different Business Object ID or including information about related Business Objects.

## Practice Exercise: Get Quick Search Configuration for Business Objects

This is the third in a series of practice exercises to help you become familiar with Cherwell's REST API Discovery Tool. We highly recommend that you complete the following exercises before you begin:

- [Practice Exercise: Get Search Items](#)
- [Practice Exercise: Get Business Object Schema](#)

In this exercise, you will use a Model Schema to build a basic request parameter, which will be passed into the API call.

1. Log into the REST API Discovery Tool. See [Opening the REST API Discovery Tool](#).
2. Click the **Searches** heading to expand the list of operations.
3. Click `/api/V1/getquicksearchconfigurationforbusobs` to expand the operation's details.
4. Under Data Type, ensure **Model Schema** is selected. Click the box under Model Schema to set its contents as the parameter value.  
The request Value field should match the Model Schema.
5. In the request Value field, replace the word string with the Business Object ID for Incident.

### Parameters

Parameter	Value
<b>request</b>	<pre style="font-family: monospace; margin: 0;">{   "busObIds": [     "6dd53665c0c24cab86870a21cf6434 ae"   ] }</pre>

Parameter content type:

For the procedure to get a Business Object ID, see [Practice Exercise: Get Business Object Schema](#).

6. Click **Try it out!**  
If the response data does not appear, ensure that the authentication button has been switched to **On**.
7. **Optional:** Try calling the operation again using a different Business Object ID.

## Practice Exercise: Save User

This is the fourth in a series of practice exercises to help you become familiar with Cherwell's REST API Discovery Tool. We highly recommend that you complete the following exercises before you begin:

- [Practice Exercise: Get Search Items](#)
- [Practice Exercise: Get Business Object Schema](#)
- [Practice Exercise: Get Quick Search Configuration for Business Objects](#)

In this exercise, you will use a Model Schema to build a complex request parameter, which will be passed into the API call in order to create a new User.

1. Log into the REST API Discovery Tool. See [Opening the REST API Discovery Tool](#).
2. Click the **Users** heading to expand the list of operations.
3. Click **/api/V2/saveuser** to expand the operation's details.
4. Set the contents of the request Value field to match the model schema.

In the next steps you will edit the JSON code as you did for [Practice Exercise: Get Quick Search Configuration for Business Objects](#). You can edit the code directly in Swagger, or you can copy the code into a text editor so that it is easier to work with.

5. Set the value for accountLocked to **false**.
6. Get the Business Object ID for UserInfo.  
For the procedure to get a Business Object ID, see [Practice Exercise: Get Business Object Schema](#).
7. Set the value for busObId to the Business Object ID for UserInfo
8. Delete the key-value pairs for busObPublicId and busObRecId from the JSON code. These values are not required for the creation of a new User.
9. Set the value for displayName to **Test User**.
10. Set the value for ldapRequired to **false**.
11. Set the value for loginId to **Test**.
12. Set the value for nextPasswordResetDate to **null**.
13. Set the value for password to **P@ssword**. Ensure this value meets the password complexity requirements set in CSM.
14. Ensure the value for passwordNeverExpires is set to **true**.
15. Set the value for securityGroupId to the Security Group ID for the User's Security Group. For this exercise, choose any Security Group.  
Call **/api/V2/getsecuritygroups** to see a list of Security Groups and their IDs.
16. Ensure the value for userCannotChangePassword is set to **true**.
17. Set the values for userInfoFields. Call **/api/V1/getbusinessobjectschema/busobid/{busobid}** to get the schema for the UserInfo Business Object. For each field:
  - a. Ensure the value for dirty is set to **true**.  
This setting must be enabled for each field in order to save the field values.

- b. Set the value for displayName to the Display Name.
  - c. Set the value for fieldId to the Field ID.
  - d. Set the value for name to the Name.
  - e. Set the value for the field.
18. Set the value for userMustChangePasswordAtNextLogin to **false**.
19. Delete the key-value pair for windowsUserId. You are creating a Cherwell User, so this value is unnecessary. See the example below of the full request body:

```
{
  "accountLocked": false,
  "bus0bId": "9338216b3c549b75607cf54667a4e67d1f644d9fed",
  "displayName": "Test User",
  "ldapRequired": false,
  "loginId": "Test",
  "nextPasswordResetDate": "null",
  "password": "Password",
  "passwordNeverExpires": true,
  "securityGroupId": "93d5aa70c8893014a2b7dc4733953c6185777e92bf",
  "userCannotChangePassword": true,
  "userInfoFields": [
    {
      "dirty": true,
      "displayName": "Full Name",
      "fieldId": "B0:9338216b3c549b75607cf54667a4e67d1f644d9fed,FI:93382178280a07634f62d74fc4bc587e3b3f479776",
      "name": "FullName",
      "value": "Test User"
    },
    {
      "dirty": true,
      "displayName": "Email",
      "fieldId": "B0:9338216b3c549b75607cf54667a4e67d1f644d9fed,FI:933821793f43a638cf23e34723b907956d324ad303",
      "name": "Email",

```

```
    "value": "user@domain.com"
  }
],
"userMustChangePasswordAtNextLogin": false,
}
```

20. Click **Try it out!**

Pay attention to the HTTP response codes you receive, and troubleshoot as appropriate:

200	Indicates a successful call.
400	Indicates a bad request due to missing parameters or incorrect formatting.
402	Indicates a failed call. The request body is formatted correctly, but the system is unable to execute the request.
404	Indicates a failed call due to a nonexistent URL.

21. **Optional:** Try adding or updating field values for the User you just created.

## Securing the REST API

CSM supports the OAuth2 protocol to authenticate and authorize calls to the REST API. The OAuth2 protocol enables third-party clients to obtain access to HTTP services. In basic OAuth2 message flow, the client interacts with a resource server and an authorization server. The client requests tokens from the authorization server on behalf of the user. The authorization server authenticates the User and returns tokens to the client. The client then sends the tokens to the resource server, which hosts the protected resource, and the User is able to access the resource as long as the tokens remain valid. In our environment, CSM acts as both the resource server and the authorization server.

You may optionally configure your system to support the SAML authentication protocol for the REST API. When SAML authentication is configured, CSM no longer acts as the authorization server in the OAuth2 message flow. Instead, your SAML identity provider acts as the authorization server. The identity provider receives token requests from the client, authenticates Users, and returns tokens to authorize access to the Cherwell REST API.

# OAuth2 Protocol for the REST API

All clients follow a basic message flow to access the Cherwell REST API using OAuth2. To begin, a user must [obtain a client ID](#) from CSM Administrator. Client IDs contribute to the security of the REST API by providing unique keys that work in conjunction with a User's CSM privileges.

After obtaining the client ID, the User performs a full login to the REST API using the client ID and their CSM User credentials. A successful login generates an access token, which allows the User agent to access the REST API as long as the token remains valid.



**Note:** The security settings that are configured for your Users will remain in effect when they access the REST API.

An Access token is needed for all subsequent calls to the REST API to identify the User agent as a valid API user. The life span of the Access token is based on the client ID's settings. Access tokens tend to have a relatively short life span and can be refreshed with a Refresh token.

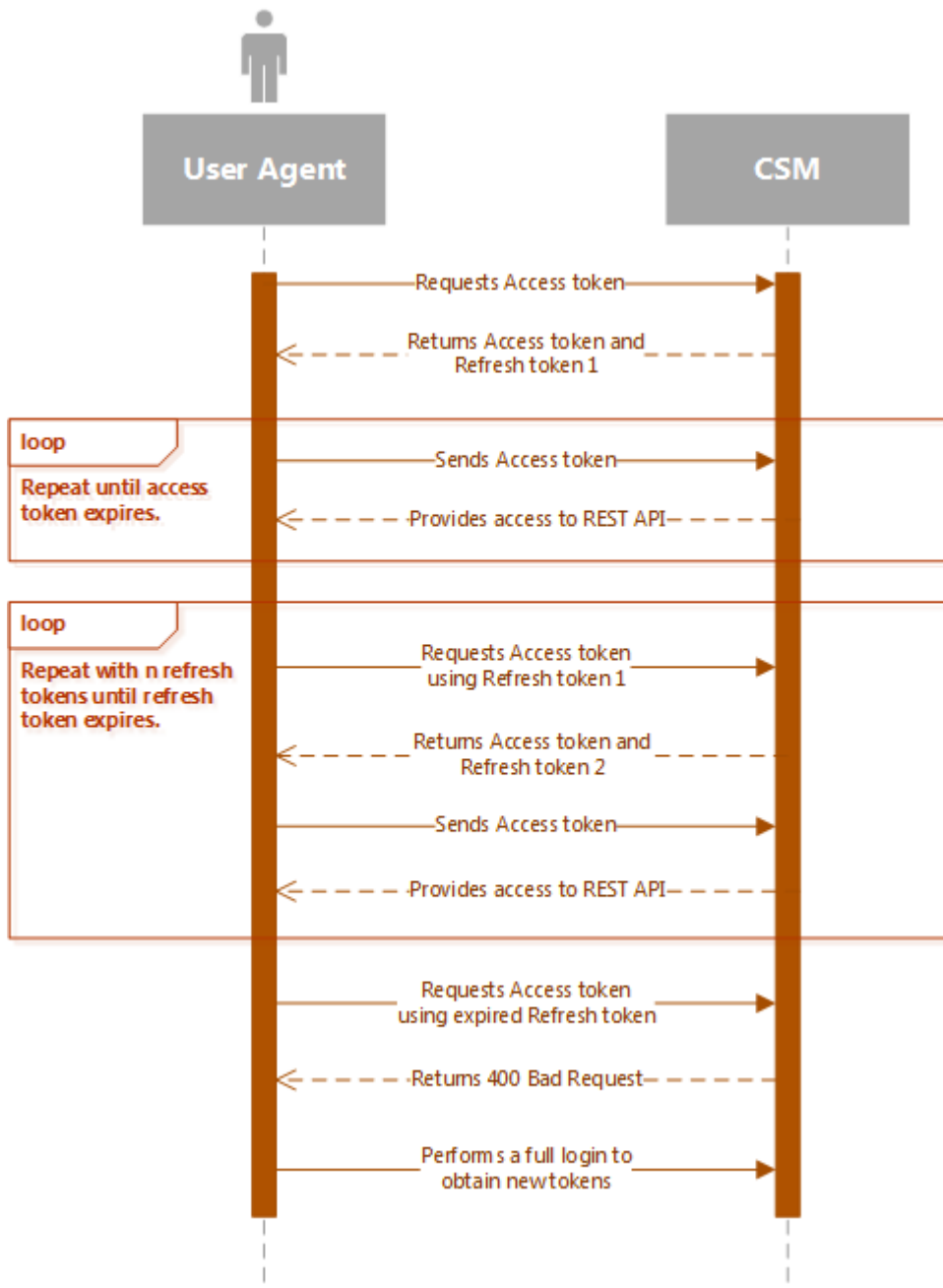
Refresh tokens are used to periodically refresh the Access token without the need to provide credentials again. Refresh tokens tend to have a longer life span than Access tokens and are also based on the client ID's settings. Once the Refresh token has expired or the user is logged out, a full login must be performed to obtain new tokens. It is not always necessary to use Refresh tokens, which are generally considered safer over non-encrypted transport protocols, such as HTTP. If you use an HTTPS connection, you can increase the Access token life span and use Access tokens for all subsequent connections.



**Note:** To further secure your system, use SSL.

Users who log in to the REST API client consume a CSM license.





## Obtaining API Client IDs

Client IDs are created in the CSM Administrator.



**Note:** You can create separate client IDs to control access for specific Users and specific integration tools.

1. From the Main window, select **Security**.
2. Click **Edit REST API client settings**.
3. Click the **Plus** icon.
4. Provide these settings for the client ID:

Setting	Description
Name	Provide a name for the client ID.
Culture	Select a language-specific culture.
Token lifespan	Set the amount of time the access token will be active.
Refresh Token lifespan	Set the amount of time the refresh token will be active.
API access is enabled	When selected, the client ID is enabled. When cleared, the client ID is disabled. Clear the check box to disable the client ID without deleting it.
Allow anonymous access	Select to make the REST API available to anonymous users.

5. Copy the client ID and provide it to REST API users.

# OAuth2 Authentication Modes

The REST API uses the authentication types specified for the CSM Browser Client:

- **Internal**

Uses the login ID and password specified for a user in CSM. If no other mode is specified, Internal mode is used.

- **LDAP**

Uses the LDAP settings configured for CSM and the server variable LOGON\_USER to attempt to find a CSM user. You can also use domain\username and password.

- **SAML**

Uses the SAML settings configured for CSM to validate credentials and find the CSM user.

For SAML configuration steps, refer to [Configuring the REST API for SAML Authentication](#).

- **Windows**

Uses the server variable LOGON\_USER to attempt to find a CSM user. You can also use domain\username and password.

Make an HTTP POST call to the *token* operation to get an access token. Pass data into the request body according to a specific authentication type. CSM returns a JSON response that includes information about the access token. The example below shows the response body for the internal authentication mode:

```
{
  "access_token": "SampleAccessTokenValue",
  "token_type": "bearer",
  "expires_in": 1199,
  "refresh_token": "SampleRefreshTokenValue",
  "as:client_id": "SampleClientIdValue",
  "username": "Username",
  ".issued": "Fri, 31 Mar 2017 15:31:39 GMT",
  ".expires": "Fri, 31 Mar 2017 15:51:39 GMT"
}
```

# Authenticating Using the Internal Mode

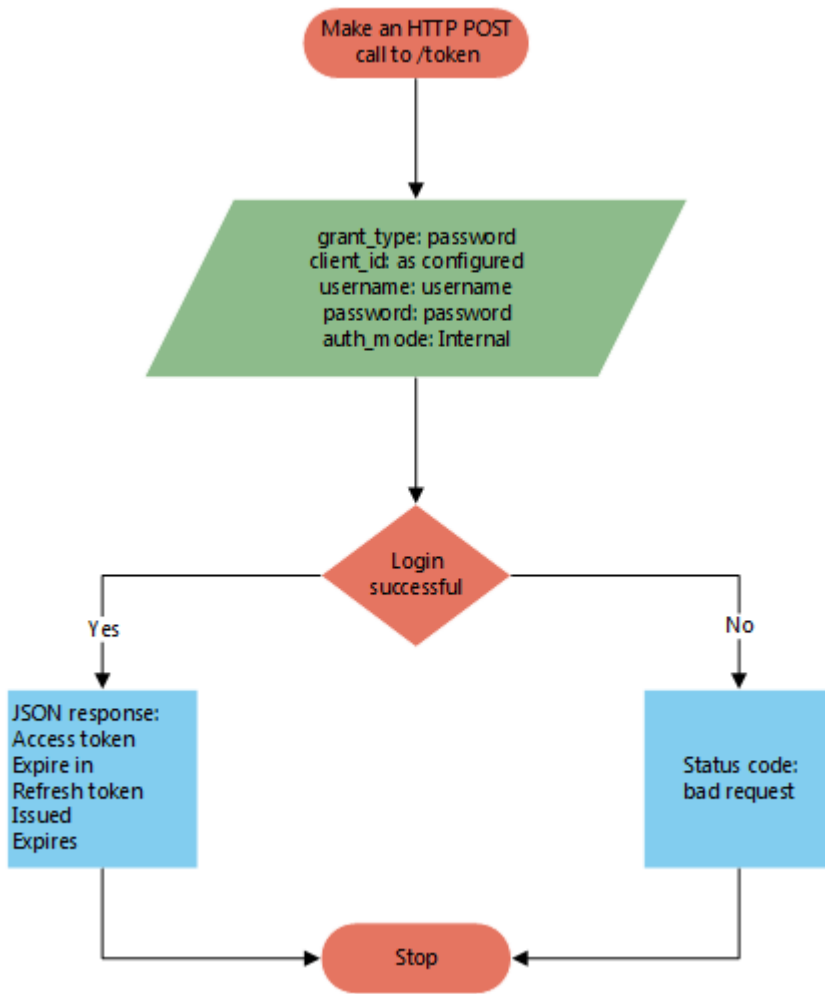
In this scenario, the User logs in to the REST API using CSM credentials.

1. Make an HTTP POST call to the token operation.
2. Pass data into the request body:

Parameter	Value
grant_type	password
client_id	as configured
username	username
password	password
auth_mode	internal

3. CSM attempts to log in the User using the supplied credentials.

If the login is successful, an Access token and Refresh token are granted. If the login is unsuccessful, an error is returned.



# Authenticating Using the Client-Based Windows/LDAP Mode

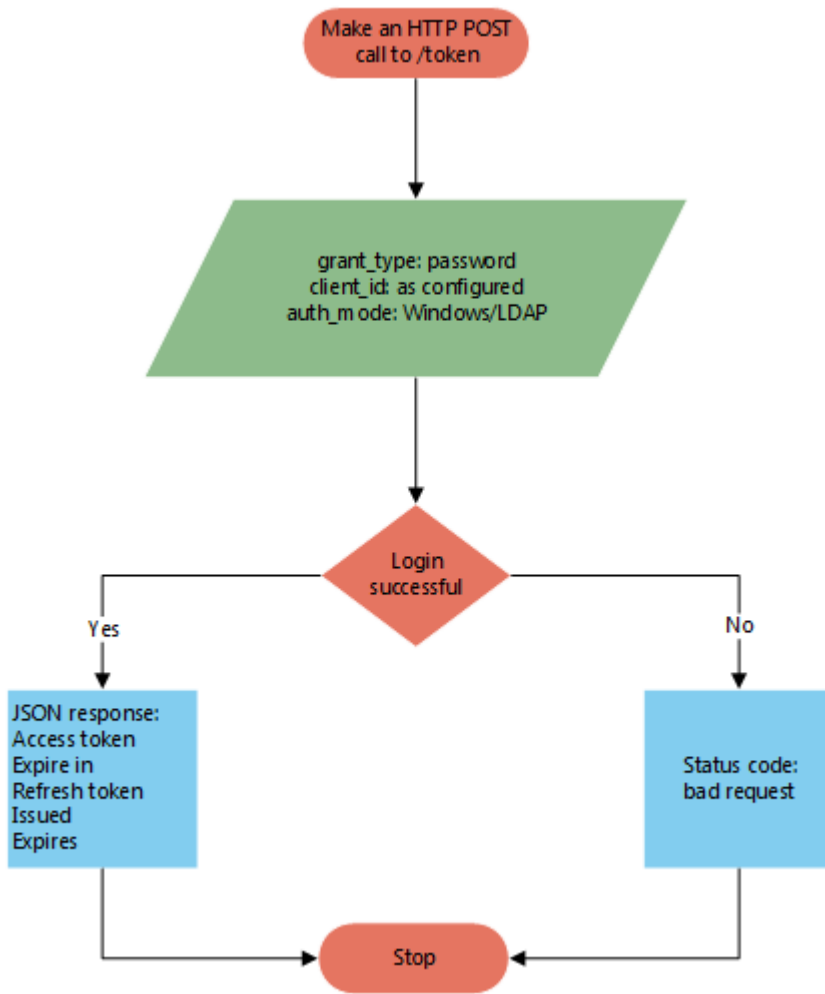
In this scenario, the User is already logged in to the Internet Information Services (IIS) manager using Windows credentials. CSM attempts to leverage the Windows ID and password to log in the User to the REST API so that entering additional credentials is not required. A successful login means the same User is logged in to both the operating system and the REST API.

1. Make an HTTP POST call to the token operation.
2. Pass data into the request body:

Parameter	Value
grant_type	password
client_id	as configured
auth_mode	Windows/LDAP

3. CSM attempts to log in the User using the Windows credentials leveraged from IIS.

If the login is successful, an Access token and Refresh token are granted. If the login is unsuccessful, an error is returned.



# Authenticating Using the Windows/LDAP Mode With Entered Credentials

In this scenario, the User overrides CSM's attempt to log in using Windows user credentials from the IIS manager by passing in alternative Windows credentials. A successful login means a different user is logged in to the REST API than the operating system.

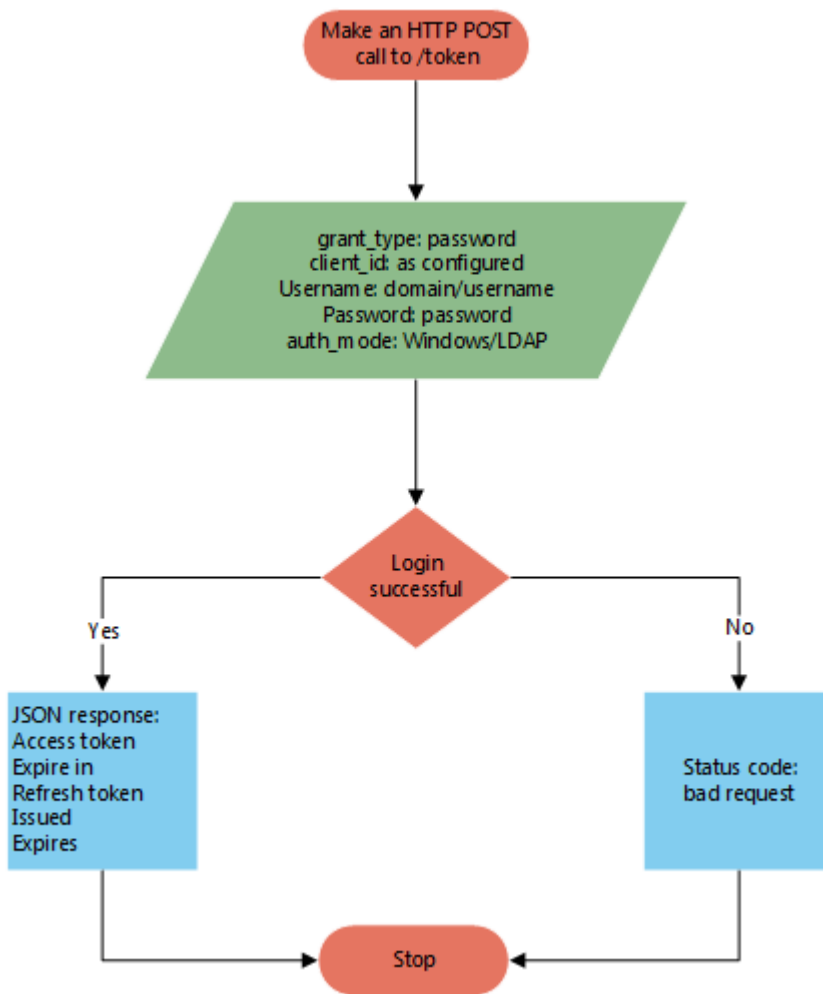
1. Make an HTTP POST call to the token operation.
2. Pass data into the request body:

Parameter	Value
grant_type	password
client_id	as configured
username	domain/username
password	password
auth_mode	Windows/LDAP

3. CSM attempts to log in the User using the supplied credentials.

If the login is successful, an Access token and Refresh token are granted. If the login is unsuccessful, an error is returned.





## Requesting a Refresh Token

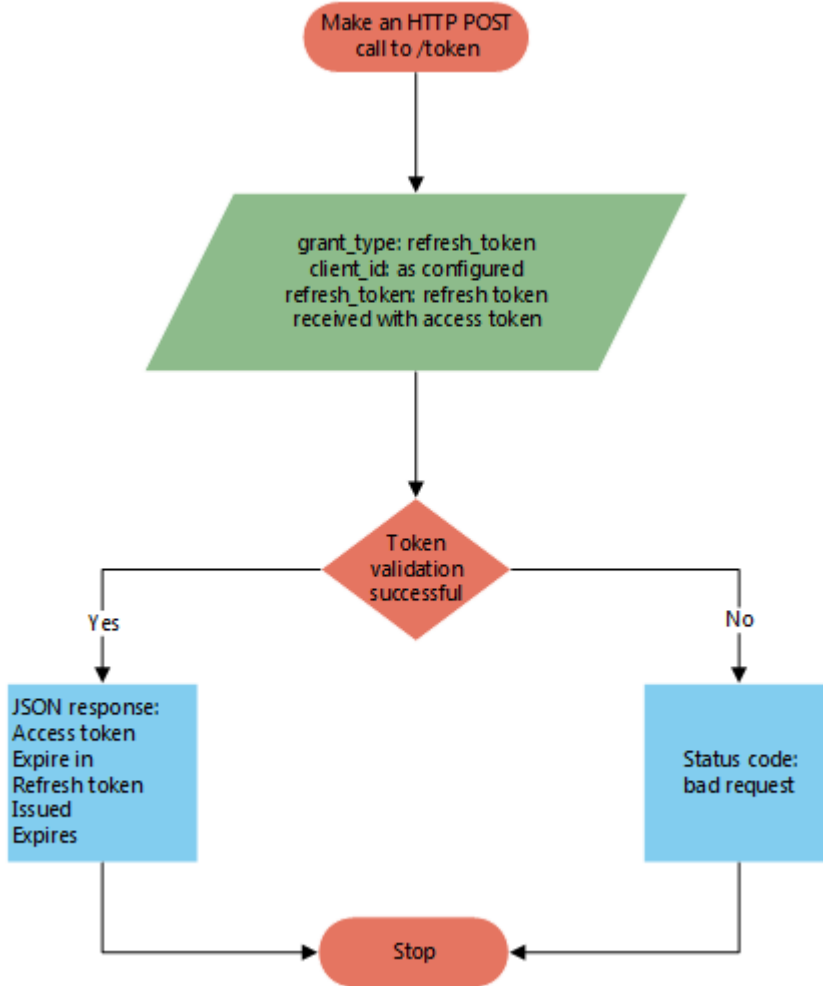
In this scenario, CSM returns a new Refresh token and invalidates the Refresh token that was previously in use.

1. Make an HTTP POST call to the token operation.
2. Pass data into the request body:

Parameter	Value
grant_type	refresh_token
client_id	as configured
refresh_token	refresh token received with access token

3. CSM validates the client ID and Refresh token using the token store.

If the validation is successful, an Access token and new Refresh token are returned. If the validation is unsuccessful, an error is returned. The new Refresh token is the one that must be used for the next Refresh token request.



# SAML Protocol for the REST API

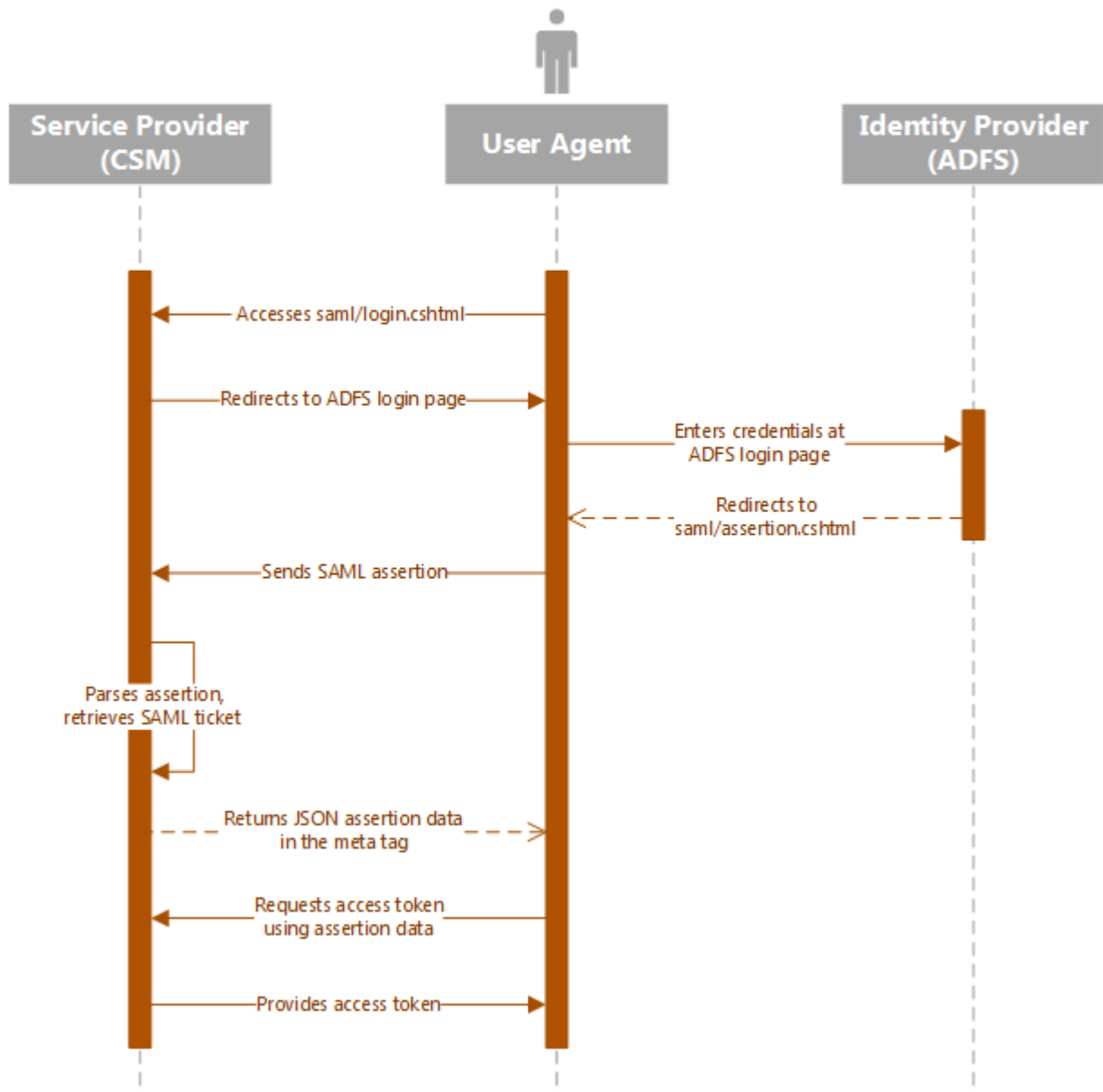
All clients follow a basic message flow to access the REST API using SAML. Whereas CSM acts as both the service provider and the identity provider in OAuth2 protocol, SAML protocol introduces a third-party identity provider.

For information about supported identity providers and the configuration procedure see [Configure the SAML Identity Provider](#).

When a User navigates to the REST API, CSM redirects the User agent to the single sign-on service at the User's identity provider where the User enters his credentials. The User's identity is authenticated by the identity provider. If the authentication is successful, the identity provider returns a SAML assertion to CSM. The assertion indicates that a trusted identity provider successfully authenticated the User so that CSM can proceed to grant access to the REST API. CSM parses the assertion and returns the assertion data to the User agent. The User agent uses the assertion data to make a request to the token operation, and passes data in the request body as shown:

Parameter	Value
grant_type	password
client_id	as usual
username	e-mail address
password	parsed SAML ticket
auth_mode	SAML

If the request is successful, CSM returns an access token to the User agent. The access token allows the User agent to access the REST API as long as the token remains valid. The following diagram shows the basic SAML message flow:



### SAML Protocol Using a Final URI Query String

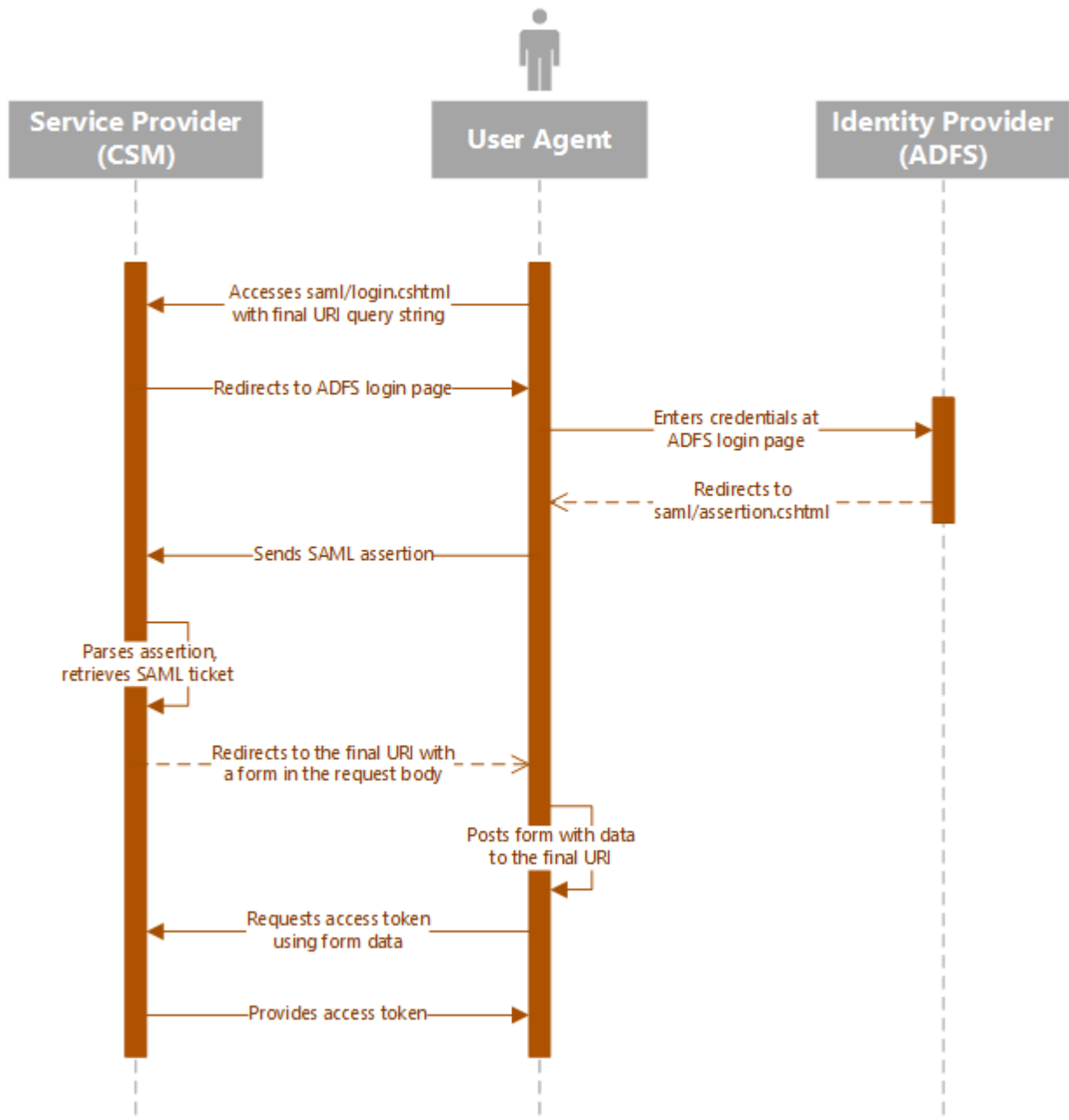
You may optionally use a final URI query string to pass form data to token requests. Example:

```
saml/login.cshhtml?finalUri=http://localhost/SamlSampleBrowserApp/default.aspx?testQueryParam=myValueForMySystem
```

After the SAML assertion is parsed, a web form is returned with an auto-submit method on the body of the form. The method posts form data to the requested final URI. Example:

```
<html>
<body onload='document.forms["samlResult"].submit() '>
<form name='samlResult' action='{finalUri}' method='post'>
<input type='hidden' name='userId' value='{nameId}'>
<input type='hidden' name='nameQualifier' value='{nameQualifier}'>
<input type='hidden' name='ticket' value='{ticket}'>
<input type='hidden' name='result' value='ok'>
<input type='hidden' name='statusCode' value='{statusCode}'>
<input type='hidden' name='statusMessage' value='{statusMessage}'>
</form>
</body>
</html>
```

The following diagram shows the SAML message flow with a final URI:



# Configuring the REST API for SAML Authentication

To use SAML authentication with the REST API, you must add the CSM base URL to the web.config file on the server that hosts your CSM Web Applications.

If you use a server farm, these steps must be completed on each server.

## To configure the REST API web.config file:

1. Navigate to \Cherwell Browser Applications\CherwellAPI.
2. Edit the Web.config file.
3. Find this key:

```
<add key="SamlBaseUri" value="place your value here" >
```

4. Change the value to the base URL for the REST API Web domain. For example:

```
https://hostname/CherwellAPI/
```

5. Save your changes.
6. Restart Internet Information Services (IIS).



## Client Example: No Final URI

The following shows how to create a client for handling SAML protocol without a final URI:

```
webBrowser.Navigate("http://host/CherwellApi/saml/login.cshtml");

private void OnNavigated(object sender, NavigationEventArgs e)
{
    dynamic doc = webBrowser.Document;

    // Look for SAMLPOSTBACK meta tag - this is how we know we have the correct page
    var elements = doc.GetElementsByTagName("meta");
    foreach (HTMLMetaElement element in elements)
    {
        if (string.Compare(element.name, "SAMLPOSTBACK", StringComparison.OrdinalIgnoreCase) == 0)
        {
            // if this were a popup dialog - could hide the dialog here and just return the content to a calling page

            // the json data is stored in the metadata
            string samlAssertionData = element.content;
            LoginToRest(samlAssertionData);
        }
    }
}

private void LoginToRest(string samlAssertionData)
{
    var samlAssertion = JObject.Parse(samlAssertionData);
    // Setup the default header
    var baseuri = "http://host/CherwellApi/";
    var apiversion = "1";
    var apiClient = new ApiClient(baseuri);
}
```

```
string clientid = "your client id";
string user = samlAssertion.GetValue("NameId", StringComparison.InvariantCultureIgnoreCase).ToString();
string password = samlAssertion.GetValue("Ticket", StringComparison.InvariantCultureIgnoreCase).ToString();

try
{
    // Login to REST service
    ServiceApi service = new ServiceApi(apiClient);
    TokenResponse tokenResponse = service.ServiceToken("password", clientid, user, password, string.Empty, "saml");
    // Setup the default header
    apiClient.DefaultHeader.Add("Authorization", "Bearer " + tokenResponse.AccessToken);

    // Create search results request
    SearchResultsRequest searchResults = new SearchResultsRequest { BusObjId = "6dd53665c0c24cab86870a21cf6434ae" };

    // Query for data
    SearchesApi searches = new SearchesApi(apiClient);
    SearchResultsResponse response = searches.SearchesGetSearchResultsAdHocV1(searchResults);
    string message = "There are " + response.TotalRows + " incidents";
    MessageBox.Show(message);
}
catch (ApiException ex)
{
    MessageBox.Show(ex.Message);
}
}
```



## Client Example: With Final URI

The following shows how to create a client for handling SAML protocol with a final URI:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Request.Form.HasKeys() && Request.Form.Keys.Cast<string>().Any(x => x
    == "ticket"))
    {
        var userId = Request.Form["userId"];
        var nameQualifier = Request.Form["nameQualifier"];
        var ticket = Request.Form["ticket"];
        var result = Request.Form["result"];
        var statusCode = Request.Form["statusCode"];
        var statusSubCode = Request.Form["statusSubCode"];
        var statusMessage = Request.Form["statusMessage"];

        var clientId = ConfigurationManager.AppSettings["clientId"];
        string baseuri = ConfigurationManager.AppSettings["apiBaseUri"];

        // Login to CherwellApi
        ServiceApi service = new ServiceApi(baseuri);
        TokenResponse tokenResponse = service.ServiceToken("password", clientId,
string.Empty,
                userId, ticket, string.Empty,
                "SAML");

        // Create search results request
        SearchResultsRequest searchResults = new SearchResultsRequest
        {
            BusObId = "6dd53665c0c24cab86870a21cf6434ae"
        };

        // create new instance of the Searches operations
        SearchesApi searches = new SearchesApi(baseuri);
```

```
// Setup the default header for authorization
searches.Configuration.DefaultHeader.Add("Authorization", "Bearer " + tokenResponse.AccessToken);

SearchResultsResponse response = searches.SearchesGetSearchResultsAdHocV1(searchResults);

// display results indicating success
TotalRows.Text = "There are " + response.TotalRows + " incidents";

var dict = Request.QueryString.Keys.Cast<string>().ToDictionary(key => key, key => Request.QueryString[key]);
var json = new JavaScriptSerializer().Serialize(dict);

QueryParameters.Text = json;
}
else
{
    // should be known if a user is attempting a SAML login to your system
    // make SAML login request to the CherwellApi
    Response.Redirect(ConfigurationManager.AppSettings["apiSamlLoginUri"]);
}
}
```

## Usage and Examples

The following topics provide usage and examples for specific operations.

# Business Object Usage and Examples

The Cherwell REST API supports basic CRUD (create, read, update, and delete) operations on all Business Objects.

## Usage: Field Identifiers

CSM field identifiers come in two forms:

- **Field identifiers for the main business object:**  
BO:6dd53665c0c24cab86870a21cf6434ae,FI:c1e86f31eb2c4c5f8e8615a5189e9b19
- **Field Identifiers for mapped relational fields:**  
FI:9397b3c0357e0678f9e3c94ce290e3783e4a61cab,  
BO:9337c2311b8e8044aa3e2a48c4a95a9f5555815126,  
RE:9402756722d89a20ab008c41cc882fd219da37dc5f

Once a field has been created, its ID will not change.

## Usage: Batch Operations

You can perform batch CRUD operations that allow you build up a collection of SaveRequests, ReadRequests or DeleteRequests on the client, and then send them as a batch to be processed on the server. This can greatly improve performance.

As batch operations are being processed, if one item in the batch fails to save, the entire batch stops processing if StopOnError is set to true. If StopOnError is set to false, the system will continue processing the remaining items in the batch.

When the system finishes processing the batch, a batch response is returned to the client. Inside of that object there is a collection of responses so you can determine if each batch operation failed or succeeded.

## Usage: Creating and Updating Business Objects

Use these operations to create or update Business Objects:

- savebusinessobject
- savebusinessobjectbatch
- saverelatedbusinessobjects

To create Business Objects, do not include a record ID or public ID in the request.

To update existing Business Objects, specify the record ID or public ID in the request. If duplicate public IDs exist, you must use the record ID.

## C# Example: Get an Incident

The following example shows how to get an Incident using C#.



**Important:** Run Swagger Code Generation before attempting to modify the code so that you do not receive reference errors. Refer to [Using Swagger Code Generation](#) for the procedure.

```
using IO.Swagger.Api;

namespace Trebuchet.WebApi.IntegrationTests.ExamplesForCustomers.GetanObject
{
    public class GetAnObject
    {
        public void GetABusinessObject()
        {
            //Get an access token using CSM credentials
            var serviceApi = new ServiceApi("http://your server/CherwellApi/");

            var tokenResponse = serviceApi.ServiceToken("password",
                "your client id", null, "CSDAdmin", "CSDAdmin", null,
                "Internal");

            //Create a new Business Object api object and add the default
            header
            var businessObjectApi = new BusinessObjectApi("http://your server/CherwellApi/");
            businessObjectApi.Configuration.AddDefaultHeader("Authorization",
                "Bearer " + tokenResponse.AccessToken);

            //Get the Business Object summary by name. This returns the Business Object ID
            var businessObjectSummaryByName =
                businessObjectApi.BusinessObjectGetBusinessObjectSummaryBy
```



```
NameV1("Incident");

        //Get the Business Object by public ID
        var readResponse =
            businessObjectApi.BusinessObjectGetBusinessObjectByPublicI
dV1(
            businessObjectSummaryByName[0].BusObId, "102667");
        }
    }
}
```

## PowerShell Example: Get an Incident

The following example shows how to get an Incident using PowerShell.

```
# Set server login variables
$serverName = "serverName"
$apiKey = "your client id"
$username = "CSDAdmin"
$password = "CSDAdmin"
$baseUri = "http://${serverName}/CherwellAPI/"

# Get an access token
$tokenUri = $baseUri + "token"
$authMode = "Internal"
$tokenRequestBody =
@{
    "Accept" = "application/json";
    "grant_type" = "password";
    "client_id" = $apiKey;
    "username" = $username;
    "password" = $password
}
$tokenResponse = Invoke-RestMethod -Method POST
    -Uri "${tokenUri}?auth_mode=${authMode}&api_key=${apiKey}" -Body $tokenRe
questBody

$requestHeader = @{ Authorization = "Bearer $($tokenResponse.access_token)
" }

# Get the business object summary for Incident. This will give us the busO
bId
$summaryUri = $baseUri + "api/v1/getbusinessobjectsummary/busobname/Incide
nt"
$summaryResults = Invoke-RestMethod -Method GET -Uri $summaryUri -ContentT
ype application/json
```

```
-Header $requestHeader
$busObId = $summaryResults[0].busobId

# Get the business object by publicId
$getIncidentUri = $baseUri + "api/V1/getbusinessobject/busobid/{busobid}/
publicid/102522"
```

## C# Example: Create an Incident

The following example shows how to create an Incident using C#.



**Important:** Run Swagger Code Generation before attempting to modify the code so that you do not receive reference errors. Refer to [Using Swagger Code Generation](#) for the procedure.

```
using System.Collections.Generic;
using System.Linq;
using IO.Swagger.Api;
using IO.Swagger.Model;

namespace Trebuchet.WebApi.IntegrationTests.ExamplesForCustomers.Create_an
_Object
{
    public class CreateAnObject
    {
        public void CreateAnIncident()
        {
            //Get an access token using CSM credentials
            var serviceApi = new ServiceApi("http://your server/CherwellAp
i/");

            var tokenResponse = serviceApi.ServiceToken("password",
                "your client id", null, "CSDAdmin", "CSDAdmin", null,
                "Internal");

            //Create a new Business Object api object and add the default
header
            var businessObjectApi = new BusinessObjectApi("http://your ser
ver/CherwellApi/");
            businessObjectApi.Configuration.AddDefaultHeader("Authorization
",
                "Bearer " + tokenResponse.AccessToken);

            //Create a new Searches api object and add the default header
```

```
var searchesApi = new SearchesApi("http://your server/Cherwell
Api");

searchesApi.Configuration.AddDefaultHeader("Authorization",
    "Bearer " + tokenResponse.AccessToken);

//Get the Business Object summary for customer internal
var businessObjectSummaryCustomerInternal =
    businessObjectApi.BusinessObjectGetBusinessObjectSummaryBy
NameV1("CustomerInternal");

//Get the Business Object schema for customer internal
var schemaResponse =
    businessObjectApi.BusinessObjectGetBusinessObjectSchemaV1(
    businessObjectSummaryCustomerInternal[0].BusObId, fals
e);

//Create the Search results request to lookup the customer and
get the customers record ID
var searchResultsRequest = new SearchResultsRequest();
searchResultsRequest.BusObId = businessObjectSummaryCustomerIn
ternal[0].BusObId;
searchResultsRequest.Filters = new List<FilterInfo>();
var filterInfo = new FilterInfo();
filterInfo.FieldId =
    schemaResponse.FieldDefinitions.First(f => f.Name == "FullN
ame").FieldId;
filterInfo.Operator = "eq";
filterInfo.Value = "Eric Cox";
searchResultsRequest.Filters.Add(filterInfo);

//Run the Search
var searchResultsResponse =
    searchesApi.SearchesGetSearchResultsAdHocV1(searchResultsR
equest);
```

```
//Set the record ID to be used in the creation of the Incident
var customerRecId = searchResultsResponse.BusinessObjects[0].Bus
usObRecId;

//Get the field template for Incident to help set the fields
var templateRequest = new TemplateRequest();

//Get the Business Object summary for Incident
var businessObjectSummaryIncident =
    businessObjectApi.BusinessObjectGetBusinessObjectSummaryBy
NameV1("Incident");

templateRequest.BusObId = businessObjectSummaryIncident[0].Bus
ObId;

templateRequest.IncludeAll = true;

//Use the template to set the fields
var templateResponse =
    businessObjectApi.BusinessObjectGetBusinessObjectTemplateV
1(templateRequest);

SetFieldValue(templateResponse.Fields, "Status", "New");
SetFieldValue(templateResponse.Fields, "Description", "New Inc
ident");
SetFieldValue(templateResponse.Fields, "ShortDescription", "Sh
ort Description");
SetFieldValue(templateResponse.Fields, "CustomerRecID", custom
erRecId);
SetFieldValue(templateResponse.Fields, "Priority", "2");
SetFieldValue(templateResponse.Fields, "Source", "Phone");
SetFieldValue(templateResponse.Fields, "IncidentType", "Incide
nt");
```

```
        SetFieldValue(templateResponse.Fields, "Service", "Employee Support");
        SetFieldValue(templateResponse.Fields, "Category", "Add/Change");
        SetFieldValue(templateResponse.Fields, "Subcategory", "New Employee Setup");

        //Create the save request
        var saveRequest = new SaveRequest();
        saveRequest.BusObjId = businessObjectSummaryIncident[0].BusObjId;

        saveRequest.Fields = templateResponse.Fields;

        //Create the Incident
        var saveResponse = businessObjectApi.BusinessObjectSaveBusinessObjectV1(saveRequest);
    }

    public void SetFieldValue(List<FieldTemplateItem> fields, string fieldName,
        string fieldValue)
    {
        var fieldTemplate = fields.First(s => s.Name.Equals(fieldName));
        if (fieldTemplate != null)
        {
            fieldTemplate.Value = fieldValue;
            fieldTemplate.Dirty = true;
        }
    }
}
```

## PowerShell Example: Create an Incident

The following example shows how to create an Incident using PowerShell.

```
#-----  
-----  
# Functions  
#-----  
-----  
  
Function Set-FieldValue  
{  
    [CmdletBinding()]  
    Param(  
        [Parameter(Position=0, Mandatory=$True)]  
            [PSCustomObject]$template  
        , [Parameter(Position=0, Mandatory=$True)]  
            [String]$fieldName  
        , [Parameter(Position=0, Mandatory=$True)]  
            [String]$value  
    )  
  
    $field = $template.fields | Where-Object {$_.name -eq $fieldName}  
  
    if (!$field)  
    {  
        throw [System.Exception]"fieldName does not exist in template"  
    }  
  
    $field.value = $value  
    $field.dirty = $true  
  
}
```

```
#-----
```



```
-----  
# Logon to Service  
#-----  
-----  
  
# Set server login variables  
$serverName = "your server"  
$apiKey = "your client id"  
$userName = "CSDAdmin"  
$password = "CSDAdmin"  
$baseUri = "http://${serverName}/CherwellAPI/"  
  
# Get an access token  
$tokenUri = $baseUri + "token"  
$authMode = "Internal"  
$tokenRequestBody =  
@{  
    "Accept" = "application/json";  
    "grant_type" = "password";  
    "client_id" = $apiKey;  
    "username" = $userName;  
    "password" = $password  
}  
$tokenResponse = Invoke-RestMethod -Method POST  
    -Uri "${tokenUri}?auth_mode=${authMode}&api_key=${apiKey}" -Body $tokenRe  
questBody  
  
$requestHeader = @{ Authorization = "Bearer $($tokenResponse.access_token)  
" }  
  
#-----  
-----
```

```

# Get Customer Data
#-----
-----

# Get the business object summary for customer internal
$summaryUri = $baseUri + "api/V1/getbusinessobjectsummary/busobname/Custom
erInternal"
$summaryResults = Invoke-RestMethod -Method GET -Uri $summaryUri -ContentT
ype application/json
-Header $requestHeader
$busobId = $summaryResults[0].busobId

# Get the business object schema for customer interal
$schemaUri = $baseUri + "api/V1/getbusinessobjectschema/busobid/" + $busob
Id
$schemaResults = Invoke-RestMethod -Method GET -Uri $schemaUri -ContentTyp
e application/json
-Header $requestHeader

# Get the fieldId for the Full Name field so we can use it for a search.
$fullNameField = $schemaResults.fieldDefinitions | Where-Object {$_.displayN
ame -eq "Full name"}

# Create the search results request to lookup the customer and get the cus
tomers recid
$filterInfo =
@{
    fieldId = $($fullNameField.fieldId);
    operator = "eq";
    value = "Eric Cox"
}
$searchResultsRequest =
@{
    busObID = $busobId;

```

```
        filters = a($filterInfo)
    } | ConvertTo-Json

# Run the search
$searchUri = $baseUri + "api/V1/getsearchresults"
$searchResponse = Invoke-RestMethod -Method POST -Uri $searchUri -ContentType application/json
                -Header $requestHeader -Body $searchResultsRequest

# Set the recid to be used in the creation of the incident
$customerRecId = $searchResponse.businessObjects[0].busObRecid

# Get the business object summary for incident
$summaryUri = $baseUri + "api/V1/getbusinessobjectsummary/busobname/Incident"
$summaryResponse = Invoke-RestMethod -Method GET -Uri $summaryUri -ContentType application/json
                -Header $requestHeader
$busobId = $summaryResponse[0].busobId

#-----
#-----
# Create a Business Object Template for BusObID for the specified criteria
#-----
#-----

# Create request for the business object template POST method
$getTemplateUri = $baseUri + "api/V1/GetBusinessObjectTemplate"
$templateRequest =
@{
    busObId = $busobId;
```

```

    includeRequired = $true;
    includeAll = $true
} | ConvertTo-Json
$templateResponse = Invoke-RestMethod -Method POST -Uri $getTemplateUri -Header $requestHeader
    -ContentType application/json -Body $templateRequest

# Set values in the template
Set-FieldValue -template $templateResponse -fieldName "Status" -value "New"
Set-FieldValue -template $templateResponse -fieldName "Description" -value "From Powershell using REST API"
Set-FieldValue -template $templateResponse -fieldName "ShortDescription" -value "New Incident"
Set-FieldValue -template $templateResponse -fieldName "CustomerRecID" $customerRecId
Set-FieldValue -template $templateResponse -fieldName "Priority" -value "2"
Set-FieldValue -template $templateResponse -fieldName "Source" -value "Phone"
Set-FieldValue -template $templateResponse -fieldName "IncidentType" -value "Incident"
Set-FieldValue -template $templateResponse -fieldName "Service" -value "Employee Support"
Set-FieldValue -template $templateResponse -fieldName "Category" -value "Add/Change"
Set-FieldValue -template $templateResponse -fieldName "Subcategory" -value "New Employee Setup"

#-----
-----
# Create New Business Object
#-----
-----

```

```
# Get the fields portion of the template and use it in the request for a new BO
#createBOUri = $baseuri + "api/V1/SaveBusinessObject"
#createBORequest =
@{
    busObId = $busobId;
    fields = @($($templateResponse.fields))
} | ConvertTo-Json

# Submit business object to server
#createBOResponse = Invoke-RestMethod -Method POST -Uri $createBOUri -Header $requestHeader
-ContentType application/json -Body $createBORequest
```

## C# Example: Upload Attachments

The following example shows how to upload attachments using C#.



**Important:** Run Swagger Code Generation before attempting to modify the code so that you do not receive reference errors. Refer to [Using Swagger Code Generation](#) for the procedure.

```
using System.IO;
using IO.Swagger.Api;

namespace Trebuchet.WebApi.IntegrationTests.ExamplesForCustomers.Attachments
{
    public class UploadAttachment
    {
        public void UploadAnAttachmentByBusinessObjectIdAndRecordId()
        {
            //Get an access token using CSM credentials
            var serviceApi = new ServiceApi("http://your server/CherwellApi/");

            var tokenResponse = serviceApi.ServiceToken("password",
                "your client id", null, "CSDAdmin", "CSDAdmin", null,
                "Internal");

            //Create a new Business Object api object and add the default
            header
            var businessObjectApi = new BusinessObjectApi("http://your server/CherwellApi/");
            businessObjectApi.Configuration.AddDefaultHeader("Authorization",
                "Bearer " + tokenResponse.AccessToken);

            //Get the Business Object summary for Incident
            var businessObjectSummaryIncident =
                businessObjectApi.BusinessObjectGetBusinessObjectSummaryBy
```

```
NameV1("Incident");

    var allFileBytes =
        File.ReadAllBytes(
            "path to file");
    var totalSize = allFileBytes.Length;

    businessObjectApi.BusinessObjectUploadBusinessObjectAttachment
    ByIdAndRecIdV1(
        allFileBytes,
        "TextFile", businessObjectSummaryIncident[0].BusObjId,
        "941c01f21ee2de2f65ace74428a5f5955a3e621314", 0, totalSize
    , string.Empty,
        string.Empty);
    }
}
}
```

## PowerShell Example: Upload Attachments

The following example shows how to upload attachments using PowerShell.

```
# Set server login variables
$serverName = "your server"
$apiKey = "your client id"
$username = "CSDAdmin"
$password = "CSDAdmin"
$baseUrl = "http://${serverName}/CherwellAPI/"

# Get an access token
$tokenUri = $baseUrl + "token"
$authMode = "Internal"
$tokenRequestBody =
@{
    "Accept" = "application/json";
    "grant_type" = "password";
    "client_id" = $apiKey;
    "username" = $username;
    "password" = $password
}
$tokenResponse = Invoke-RestMethod -Method POST
-Uri "${tokenUri}?auth_mode=${authMode}&api_key=${apiKey}" -Body $tokenReq
uestBody

$requestHeader = @{ Authorization = "Bearer $($tokenResponse.access_token)
" }

# Get the business object summary for Incident. This will give us the busO
bId
$summaryUri = $baseUrl + "api/v1/getbusinessobjectsummary/busobname/Incide
nt"
$summaryResults = Invoke-RestMethod -Method GET -Uri $summaryUri -ContentT
ype application/json
```



```
-Header $requestHeader
$busObId = $summaryResults[0].busobId

# Get an incident rec id from the public id
$getIncidentUri = $baseUri + "api/V1/getbusinessobject/busobid/{busobid}/
publicid/102521"
$getIncidentResponse = Invoke-RestMethod -Method GET -Uri $getIncidentUri
-ContentType application/json -Header $requestHeader
$busObRecId = $getIncidentResponse.busObRecId

# Create a text file
"This file was attached to this incident by the API example for PowerShell"
|
Out-File -FilePath "$($env:TEMP)\test.txt"
$file = Get-ChildItem -Path "$($env:TEMP)\test.txt"
$filename = $file.Name
$totalsize = $file.Length
$allFileBytes = [System.IO.File]::ReadAllBytes($file.FullName)
$offset = 0

# Upload the file to the incident
$uploadAttachmentUri = $baseUri + "api/V1/uploadbusinessobjectattachment/fi
lename/{filename}
/busobid/{busobid}/busobrecid/{busobrecid}/offset/{offset}/totalsize/{
totalsize}"
$uploadAttachmentRequest = $allFileBytes
$uploadAttachmentResponse = Invoke-RestMethod -Method POST -Uri $uploadAtt
achmentUri
-ContentType application/json -Header $requestHeader -Body $uploadAttachm
entRequest

# Delete the text file that was created
Remove-Item $file.FullName -Force
```

## C# Example: Delete a Business Object

The following example shows how to delete a Business Object using C#.



**Important:** Run Swagger Code Generation before attempting to modify the code so that you do not receive reference errors. Refer to [Using Swagger Code Generation](#) for the procedure.

```
using IO.Swagger.Api;

namespace Trebuchet.WebApi.IntegrationTests.ExamplesForCustomers.Delete
{
    public class DeleteABusinessObject
    {
        public void DeleteABusinessObjectByPublicId()
        {
            //Get an access token using CSM credentials
            var serviceApi = new ServiceApi("http://your server/CherwellAp
i/");

            var tokenResponse = serviceApi.ServiceToken("password",
                "your client id", null, "CSDAdmin", "CSDAdmin", null,
                "Internal");

            //Create a new Business Object api object and add the default
            header
            var businessObjectApi = new BusinessObjectApi("http://your ser
ver/CherwellApi/");
            businessObjectApi.Configuration.AddDefaultHeader("Authorization
",
                "Bearer " + tokenResponse.AccessToken);

            //Get the Business Object summary by name. This returns the B
usiness Object ID
            var businessObjectSummaryByName =
                businessObjectApi.BusinessObjectGetBusinessObjectSummaryBy
NameV1("Incident");
```

```
        //Delete Business Object
        businessObjectApi.BusinessObjectDeleteBusinessObjectByPublicId
V1(
        businessObjectSummaryByName[0].BusObId, "123456");
    }
}
}
```

## PowerShell Example: Delete A Business Object

The following example shows how to delete a Business Object using PowerShell.

```
# Set server login variables
$serverName = "your server"
$apiKey = "your client id"
$username = "CSDAdmin"
$password = "CSDAdmin"
$baseUri = "http://${serverName}/CherwellAPI/"

# Get an access token
$tokenUri = $baseUri + "token"
$authMode = "Internal"
$tokenRequestBody =
@{
    "Accept" = "application/json";
    "grant_type" = "password";
    "client_id" = $apiKey;
    "username" = $username;
    "password" = $password
}
$tokenResponse = Invoke-RestMethod -Method POST
    -Uri "${tokenUri}?auth_mode=${authMode}&api_key=${apiKey}" -Body $tokenRe
questBody

$requestHeader = @{ Authorization = "Bearer $($tokenResponse.access_token)
" }

# Get the business object summary for Incident. This will give us the busO
bId
$summaryUri = $baseUri + "api/v1/getbusinessobjectsummary/busobname/Incide
nt"
$summaryResults = Invoke-RestMethod -Method GET -Uri $summaryUri -ContentT
ype application/json
```

```
-Header $requestHeader
$busObId = $summaryResults[0].busobId

# Delete business object
$deleteIncidentUri = $baseUri + "api/V1/deletebusinessobject/busobid/{bus
obid}/publicid/102522"
$deleteIncidentResponse = Invoke-RestMethod -Method DELETE -Uri $deleteInc
identUri
-ContentType application/json -Header $requestHeader
```

## Using Search Operations

Search operations support getting search data from either a saved query or through an ad-hoc approach. A ad-hoc search is not based on a existing saved search. You can specify filters, columns, general search, and sorting capabilities. At a minimum, you must specify the Business Object ID.

### Usage: Filters

To specify filtering, use the Filters member in the SearchResultsRequest. The Filters member is a collection of FilterInfo data structures. A FilterInfo contains the full field ID, operator and value.

You can specify more than one filter. If you add multiple filters for the same field ID, the result is an OR operation between those filters. If the field IDs are different, the result is an AND operation between those filters.

The following table shows the supported operators.

eq	Equals specified value
gt	Greater than specified value
lt	Less than specified value
contains	Contains specified value
startswith	Starts with specified value

The following example shows how to specify a filter on the Incident ID field to find a specific incident.

```
SearchResultsRequest searchResults = new SearchResultsRequest();
searchResults.BusObjId = "6dd53665c0c24cab86870a21cf6434ae";

// Create a filter on the incident ID field
searchResults.Filters = new System.Collections.Generic.List<FilterInfo>();

FilterInfo filter1 = new FilterInfo();
filter1.FullFieldId = "B0:6dd53665c0c24cab86870a21cf6434ae,FI:6ae282c55e8e4266ae66ffc070c17fa3";
filter1.Operator = "eq";
filter1.Value = "100216";
searchResults.Filters.Add(filter1);

SearchesApi searches = new SearchesApi(apiClient);
```

```
SearchResultsResponse response = searches.SearchesGetSearchResultsAdHocV1(
searchResults);
```

In the following example, there are two filters for the incident ID field. This means that any incidents with IncidentID = 100216 OR IncidentID=200367 will be found.

```
SearchResultsRequest searchResults = new SearchResultsRequest();
searchResults.BusObjId = "6dd53665c0c24cab86870a21cf6434ae";

// Create a filter on the incident ID field
searchResults.Filters = new System.Collections.Generic.List<FilterInfo>();

FilterInfo filter1 = new FilterInfo();
filter1.FullFieldId = "B0:6dd53665c0c24cab86870a21cf6434ae,FI:6ae282c55e8e4
266ae66ffc070c17fa3";
filter1.Operator = "eq";
filter1.Value = "100216";
searchResults.Filters.Add(filter1);

FilterInfo filter2 = new FilterInfo();
filter2.FullFieldId = "B0:6dd53665c0c24cab86870a21cf6434ae,FI:6ae282c55e8e4
266ae66ffc070c17fa3";
filter2.Operator = "eq";
filter2.Value = "200367";
searchResults.Filters.Add(filter2);

// Query for data
SearchesApi searches = new SearchesApi(apiClient);
SearchResultsResponse response = searches.SearchesGetSearchResultsAdHocV1(
searchResults);
```

### Usage: Fields

By default, if you do not specify fields, a default set of fields (defined by the default Grid definition for the Business Object) is used to determine which fields will be displayed. You can override this behavior and specify the fields to include in your results. Use the Fields collection and add the full IDs of the fields to include:

```
SearchResultsRequest searchResults = new SearchResultsRequest();
searchResults.BusObjId = "6dd53665c0c24cab86870a21cf6434ae";

searchResults.Fields= new System.Collections.Generic.List<string>();
searchResults.Fields.Add("B0:6dd53665c0c24cab86870a21cf6434ae,FI:6ae282c55e8e4266ae66ffc070c17fa3");
```

### Usage: Search Results Field Templates

Use *POST api/V1/getsearchresultsbusinessobjects* to use the powerful filtering, sorting, and field specification abilities of the search results API to produce results that are in the field template data structure syntax (see Business Objects / Field Templates).

You can then update values within that set of field template and then update those records. There are several scenarios where this might be useful. For example, you can find all incidents older than 1 year, and then set their status to closed.



## C# Example: Get Search Items

The following example shows how to get search items using C#.



**Important:** Run Swagger Code Generation before attempting to modify the code so that you do not receive reference errors. Refer to [Using Swagger Code Generation](#) for the procedure.

```
using IO.Swagger.Api;

namespace Trebuchet.WebApi.IntegrationTests.ExamplesForCustomers.Searches
{

    public class GetSearchItems
    {

        public void GetSearchItemsByAssociationAndScope()
        {
            //Get an access token using CSM credentials
            var serviceApi = new ServiceApi("http://server/CherwellApi/");
            var tokenResponse = serviceApi.ServiceToken("password", "your
client id", null,
                "CSDAdmin", "CSDAdmin", null, "Internal");

            //Create a new Searches api object and add the default header
            var searchesApi = new SearchesApi("http://server/CherwellApi/"
);

            searchesApi.Configuration.AddDefaultHeader("Authorization",
                "Bearer " + tokenResponse.AccessToken);

            //Create a new Business Object api object and add the default
header
            var businessObjectApi = new BusinessObjectApi("http://server/C
herwellApi/");
            businessObjectApi.Configuration.AddDefaultHeader("Authorization
```

```
    ",
        "Bearer " + tokenResponse.AccessToken);

    //Get the Business Object summary by name. This returns the B
business Object ID
    var businessObjectSummaryByName =
        businessObjectApi.BusinessObjectGetBusinessObjectSummaryBy
NameV1("Incident");

    var searchItemResponse =
        searchesApi.SearchesGetSearchItemsByAssociationScopeV1(
            businessObjectSummaryByName[0].BusObId, "Global", null
);
    }
}
}
```

## PowerShell Example: Get Search Items

The following example shows how to get search items using PowerShell.

```
# Set server login variables
$serverName = "your server"
$apiKey = "your client id"
$username = "CSDAdmin"
$password = "CSDAdmin"
$baseUrl = "http://${serverName}/CherwellAPI/"

# Get an access token
$tokenUri = $baseUrl + "token"
$authMode = "Internal"
$tokenRequestBody =
@{
    "Accept" = "application/json";
    "grant_type" = "password";
    "client_id" = $apiKey;
    "username" = $username;
    "password" = $password
}
$tokenResponse = Invoke-RestMethod -Method POST
    -Uri "${tokenUri}?auth_mode=${authMode}&api_key=${apiKey}" -Body $tokenRe
questBody

$requestHeader = @{ Authorization = "Bearer $($tokenResponse.access_token)
" }

# Get the business object summary for Incident. This will give us the busO
bId
$summaryUri = $baseUrl + "api/v1/getbusinessobjectsummary/busobname/Incide
nt"
$summaryResults = Invoke-RestMethod -Method GET -Uri $summaryUri -ContentT
ype application/json
```

```
-Header $requestHeader
$busObId = $summaryResults[0].busobId

# Get the search items (saved queries) with association of Incident from t
he Global scope
$searchUri = $baseUri + "api/V1/getsearchitems/association/${busObId}/scop
e/Global"
$searchResponse = Invoke-RestMethod -Method GET -Uri $searchUri -ContentTy
pe application/json
-Header $requestHeader
```

## C# Example: Perform an Ad-hoc Search with Filter

The following example shows how to perform an ad-hoc search for Incidents using a filter.



**Important:** Run Swagger Code Generation before attempting to modify the code so that you do not receive reference errors. Refer to [Using Swagger Code Generation](#) for the procedure.

```
using System.Collections.Generic;
using System.Linq;
using IO.Swagger.Api;
using IO.Swagger.Model;

namespace Trebuchet.WebApi.IntegrationTests.ExamplesForCustomers.Searches
{
    public class AdHocSearchWithFilter
    {
        public void SearchResultsWithFilter()
        {
            //Get an access token using CSM credentials
            var serviceApi = new ServiceApi("http://your server/CherwellAp
i/");

            var tokenResponse = serviceApi.ServiceToken("password",
                "your client id", null, "CSDAdmin", "CSDAdmin", null,
                "Internal");

            //Create a new Business Object api object and add the default
header
            var businessObjectApi = new BusinessObjectApi("http://your ser
ver/CherwellApi/");
            businessObjectApi.Configuration.AddDefaultHeader("Authorization
",
                "Bearer " + tokenResponse.AccessToken);

            //Create a new Searches api object and add the default header
            var searchesApi = new SearchesApi("http://your server/Cherwell
```

```
Api/");
    searchesApi.Configuration.AddDefaultHeader("Authorization",
        "Bearer " + tokenResponse.AccessToken);

    //Get the Business Object summary by name. This returns the B
business Object ID
    var businessObjectSummaryByName =
        businessObjectApi.BusinessObjectGetBusinessObjectSummaryBy
NameV1("Incident");

    //Get the Incident Business Object template
    var templateRequest = new TemplateRequest
    {
        BusObjId = businessObjectSummaryByName[0].BusObjId
    };
    var templateResponse =
        businessObjectApi.BusinessObjectGetBusinessObjectTemplateV
1(templateRequest);

    //Create Search results request
    var searchResults = new SearchResultsRequest();
    searchResults.BusObjId = businessObjectSummaryByName[0].BusObjId
;

    //Create a filter on the Incident ID field
    searchResults.Filters = new List<FilterInfo>();

    var filter = new FilterInfo();
    filter.FieldId =
        templateResponse.Fields.First(f => f.Name == "ShortDescri
ption").FieldId;
    filter.Operator = "eq";
    filter.Value = "Down System";
    searchResults.Filters.Add(filter);
```

```
        //Query for data
        var response = searchesApi.SearchesGetSearchResultsAdHocV1(sea
rchResults);
    }
}
}
```

## PowerShell Example: Perform an Ad-hoc Search with Filter

The following example shows how to perform an ad-hoc search for Incidents using a filter.

```
# Set server login variables
$serverName = "your server"
$apiKey = "your client id"
$username = "CSDAdmin"
$password = "CSDAdmin"
$baseUri = "http://${serverName}/CherwellAPI/"

# Get an access token
$tokenUri = $baseUri + "token"
$authMode = "Internal"
$tokenRequestBody =
@{
    "Accept" = "application/json";
    "grant_type" = "password";
    "client_id" = $apiKey;
    "username" = $username;
    "password" = $password
}
$tokenResponse = Invoke-RestMethod -Method POST
    -Uri "${tokenUri}?auth_mode=${authMode}&api_key=${apiKey}" -Body $tokenRe
questBody

$requestHeader = @{ Authorization = "Bearer $($tokenResponse.access_token)
" }

# Get the business object summary for Incident. This will give us the busO
bId
$summaryUri = $baseUri + "api/v1/getbusinessobjectsummary/busobname/Incide
nt"
$summaryResponse = Invoke-RestMethod -Method GET -Uri $summaryUri -Content
```



```
Type application/json
-Header $requestHeader
$busObjId = $summaryResponse[0].busobjId

# Get the business object template for Incident. We need this to get the field id for ShortDescription
$templateUri = $baseUri + "api/v1/getbusinessobjecttemplate"
$templateRequestBody =
@{
    busObjId = $busobjId
    includeAll = $true
} | ConvertTo-Json
$templateResponse = Invoke-RestMethod -Method POST -Uri $templateUri -ContentType application/json
-Header $requestHeader -Body $templateRequestBody
$shortDescField = $templateResponse.fields | Where-Object {$_.name -eq "ShortDescription"}

# Put together the request
$searchResultsRequest =
@{
    Q
    BusObjId = $busobjId;
    Filters = @(
        @{
            fieldId = $shortDescField.fieldId;
            Operator = "eq";
            Value = "Printer Issue"
        }
    )
} | ConvertTo-Json

# Run the search
$searchUri = $baseUri + "api/v1/getsearchresults"
$searchResultsResponse = Invoke-RestMethod -Method POST -Uri $searchUri
-ContentType application/json -Header $requestHeader -Body $searchResultsR
```

```
equest
```

```
Write-Host $searchResultsResponse
```

## C# Example: Perform an Ad-hoc Search with Sorting

The following example shows how to perform an ad-hoc search for Incidents sorted in ascending order.



**Important:** Run Swagger Code Generation before attempting to modify the code so that you do not receive reference errors. Refer to [Using Swagger Code Generation](#) for the procedure.

```
using System;
using System.Collections.Generic;
using System.Linq;
using IO.Swagger.Api;
using IO.Swagger.Model;

namespace Trebuchet.WebApi.IntegrationTests.ExamplesForCustomers.Searches
{
    public class AdHocSearchAscending
    {
        public void SearchResultsSortingAscendingByIncidentStatus()
        {
            //Get an access token using CSM credentials
            var serviceApi = new ServiceApi("http://your server/CherwellAp
i/");

            var tokenResponse = serviceApi.ServiceToken("password",
                "your client id", null, "CSDAdmin", "CSDAdmin", null,
                "Internal");

            //Create a new Business Object api object and add the default
header
            var businessObjectApi = new BusinessObjectApi("http://your ser
ver/CherwellApi/");
            searchesApi.Configuration.AddDefaultHeader("Authorization",
                "Bearer " + tokenResponse.AccessToken);

            //Create a new Searches api object and add the default header
```

```
var searchesApi = new SearchesApi("http://your server/Cherwell
Api/");
businessObjectApi.Configuration.AddDefaultHeader("Authorization
",
    "Bearer " + tokenResponse.AccessToken);

//Get the Business Object summary by name. This returns the B
usiness Object ID
var businessObjectSummaryByName =
    businessObjectApi.BusinessObjectGetBusinessObjectSummaryBy
NameV1("Incident");

//Get the schemasresponse for Incident to get the field ID for
Incident ID
var schemaResponse =
    businessObjectApi.BusinessObjectGetBusinessObjectSchemaV1(
        businessObjectSummaryByName[0].BusObId);

//Put together the request
var searchResultsRequest = new SearchResultsRequest
{
    BusObId = businessObjectSummaryByName[0].BusObId,
    PageSize = 200,
    Sorting = new List<SortInfo>
    {
        new SortInfo
        {
            FieldId =
                schemaResponse.FieldDefinitions.First(n => n.Na
me == "Status").FieldId,
            SortDirection = 1
        }
    }
};
```

```
        //Run the Search
        searchesApi.SearchesGetSearchResultsAdHocV1(searchResultsReque
st);
    }
}
```

## PowerShell Example: Perform an Ad-hoc Search with Sorting

The following example shows how to perform an ad-hoc search for Incidents sorted in ascending order.

```
# Set server login variables
$serverName = "your server"
$apiKey = "your client id"
$username = "CSDAdmin"
$password = "CSDAdmin"
$baseUri = "http://${serverName}/CherwellAPI/"

# Get an access token
$tokenUri = $baseUri + "token"
$authMode = "Internal"
$tokenRequestBody =
@{
    "Accept" = "application/json";
    "grant_type" = "password";
    "client_id" = $apiKey;
    "username" = $username;
    "password" = $password
}
$tokenResponse = Invoke-RestMethod -Method POST
    -Uri "${tokenUri}?auth_mode=${authMode}&api_key=${apiKey}" -Body $tokenRe
questBody

$requestHeader = @{ Authorization = "Bearer $($tokenResponse.access_token)
" }

# Get the business object summary for Incident. This will give us the busO
bId
$summaryUri = $baseUri + "api/v1/getbusinessobjectsummary/busobname/Incide
nt"
$summaryResponse = Invoke-RestMethod -Method GET -Uri $summaryUri -Content
```

```
Type application/json
-Header $requestHeader
$busObId = $summaryResponse[0].busobId

# Get the business object schema for Incident. We need this to get the field id for Status
$schemaUri = $baseUri + "api/V1/getbusinessobjectschema/busobid/" + $busobId
Invoke-WebRequest -Uri $schemaUri -ContentType application/json -Header $requestHeader
$statusField = $schemaResponse.fieldDefinitions | Where-Object {$_.name -eq "Status"}

# Put together the request
$searchResultsRequest = @{}
$searchResultsRequest["BusObId"] = $busObId;
$searchResultsRequest["PageSize"] = 200;
$searchResultsRequest["Sorting"] = @{}
$searchResultsRequest["Sorting"]["fieldId"] = $statusField.fieldId;
$searchResultsRequest["Sorting"]["sortDirection"] = 1;
$searchResultsRequest | ConvertTo-Json

# Run the search
$searchUri = $baseUri + "api/V1/getsearchresults"
$searchResultsResponse = Invoke-WebRequest -Uri $searchUri -ContentType application/json -Header $requestHeader -Body $searchResultsRequest
```

# Security Usage and Examples

Security operations enable you to perform CRUD (create, read, update and delete) operations for single User accounts or in a batch. You can also add and remove Users from Teams.

## Usage: Creating and Updating Users

Use these operations to create or update users:

- saveuser
- saveuserbatch

To create users, do not include a record ID or public ID in the request.

To update existing users, specify the record ID or public ID in the request. If duplicate public IDs exist, you must use the record ID.



## C# Example: Create a User

The following example shows how to create a CSM User account using C#.



**Important:** Run Swagger Code Generation before attempting to modify the code so that you do not receive reference errors. Refer to [Using Swagger Code Generation](#) for the procedure.

```
using System;
using System.Collections.Generic;
using System.Linq;
using IO.Swagger.Api;
using IO.Swagger.Model;

namespace Trebuchet.WebApi.IntegrationTests.ExamplesForCustomers.Create_an
_Object
{
    public class CreateUser
    {
        public void CreateNewUser()
        {
            //Get an access token using CSM credentials
            var serviceApi = new ServiceApi("http://your server/CherwellAp
i/");

            var tokenResponse = serviceApi.ServiceToken("password",
                "your client id", null, "CSDAdmin", "CSDAdmin", null,
                "Internal");

            //Create a new Business Object api object and add the default
header
            var businessObjectApi = new BusinessObjectApi("http://your ser
ver/CherwellApi/");
            businessObjectApi.Configuration.AddDefaultHeader("Authorization
",
                "Bearer " + tokenResponse.AccessToken);
```

```
//Create a new Security api object and add the default header
var securityApi = new SecurityApi("http://your server/Cherwell
Api/");

securityApi.Configuration.AddDefaultHeader("Authorization",
    "Bearer " + tokenResponse.AccessToken);

//Get the Business Object summary for user info
var businessObjectSummaryUserInfo =
    businessObjectApi.BusinessObjectGetBusinessObjectSummaryBy
NameV1("UserInfo");

//Get the Security Group information
var securityGroupResponse = securityApi.SecurityGetSecurityGro
upsV1();

//Get the template for userinfo. Use this to get field ids
var templateRequest = new TemplateRequest
{
    BusObId = businessObjectSummaryUserInfo[0].BusObId,
    IncludeAll = true
};

var templateResponse =
    businessObjectApi.BusinessObjectGetBusinessObjectTemplateV
1(templateRequest);

//Create the user save request
var userSaveRequest = new UserSaveRequest
{
    AccountLocked = false,
    BusObId = businessObjectSummaryUserInfo[0].BusObId,
    DisplayName = "Test User",
    LdapRequired = false,
```

```
LoginId = "Test",
NextPasswordResetDate = null,
Password = "Password",
PasswordNeverExpires = true,
SecurityGroupId =
    securityGroupResponse.SecurityGroups.First(f => f.GroupName == "Admin").GroupId,
UserCannotChangePassword = false,
UserMustChangePasswordAtNextLogin = false,
UserInfoFields = new List<FieldTemplateItem>
{
    new FieldTemplateItem
    {
        Dirty = true,
        Name = "FullName",
        Value = "Test User",
        FieldId = templateResponse.Fields.First(f => f.Name == "FullName").FieldId
    },
    new FieldTemplateItem
    {
        Dirty = true,
        Name = "EmployeeID",
        Value = "123456",
        FieldId = templateResponse.Fields.First(f => f.Name == "EmployeeID").FieldId
    },
    new FieldTemplateItem
    {
        Dirty = true,
        Name = "Comments",
        Value = "Created by api",
        FieldId = templateResponse.Fields.First(f => f.Name == "Comments").FieldId
    }
}
```

```
    },  
    new FieldTemplateItem  
    {  
        Dirty = true,  
        Name = "Department",  
        Value = "IT",  
        FieldId = templateResponse.Fields.First(f => f.Name == "Department").FieldId  
    },  
    new FieldTemplateItem  
    {  
        Dirty = true,  
        Name = "Office",  
        Value = "Colorado Springs",  
        FieldId = templateResponse.Fields.First(f => f.Name == "Office").FieldId  
    },  
    new FieldTemplateItem  
    {  
        Dirty = true,  
        Name = "Phone",  
        Value = "719-777-7777",  
        FieldId = templateResponse.Fields.First(f => f.Name == "Phone").FieldId  
    },  
    new FieldTemplateItem  
    {  
        Dirty = true,  
        Name = "CellPhone",  
        Value = "719-777-7778",  
        FieldId = templateResponse.Fields.First(f => f.Name == "CellPhone").FieldId  
    },  
    new FieldTemplateItem
```

```
        {
            Dirty = true,
            Name = "FirstName",
            Value = "Test",
            FieldId = templateResponse.Fields.First(f => f.Name == "FirstName").FieldId
        },
        new FieldTemplateItem
        {
            Dirty = true,
            Name = "MiddleInitial",
            Value = "C",
            FieldId =
                templateResponse.Fields.First(f => f.Name == "MiddleInitial").FieldId
        },
        new FieldTemplateItem
        {
            Dirty = true,
            Name = "LastName",
            Value = "User",
            FieldId = templateResponse.Fields.First(f => f.Name == "LastName").FieldId
        },
        new FieldTemplateItem
        {
            Dirty = true,
            Name = "HomePhone",
            Value = "719-777-7779",
            FieldId = templateResponse.Fields.First(f => f.Name == "HomePhone").FieldId
        },
        new FieldTemplateItem
        {
```

```

        Dirty = true,
        Name = "Address",
        Value = "1234 Cherwell Ave",
        FieldId = templateResponse.Fields.First(f => f.Name == "Address").FieldId
    },
    new FieldTemplateItem
    {
        Dirty = true,
        Name = "City",
        Value = "Colorado Springs",
        FieldId = templateResponse.Fields.First(f => f.Name == "City").FieldId
    },
    new FieldTemplateItem
    {
        Dirty = true,
        Name = "ProvinceStateName",
        Value = "CO",
        FieldId =
            templateResponse.Fields.First(f => f.Name == "ProvinceStateName")
                .FieldId
    },
    new FieldTemplateItem
    {
        Dirty = true,
        Name = "PostalCodeZip",
        Value = "80132",
        FieldId =
            templateResponse.Fields.First(f => f.Name == "PostalCodeZip").FieldId
    }
}

```

```
};

//Create the user
try
{
    securityApi.SecuritySaveUserV1(userSaveRequest);
}
catch (Exception ex)
{
    Assert.Fail(ex.Message);
}
}
}
```

## PowerShell Example: Create a User

The following example shows how to create a CSM User account using PowerShell.

```
# Set server login variables
$serverName = "your server"
$apiKey = "your client id"
$username = "CSDAdmin"
$password = "CSDAdmin"
$baseUrl = "http://${serverName}/CherwellAPI/"

# Get an access token
$tokenUri = $baseUrl + "token"
$authMode = "Internal"
$tokenRequestBody =
@{
    "Accept" = "application/json";
    "grant_type" = "password";
    "client_id" = $apiKey;
    "username" = $username;
    "password" = $password
}
$tokenResponse = Invoke-RestMethod -Method POST
    -Uri "${tokenUri}?auth_mode=${authMode}&api_key=${apiKey}" -Body $tokenRe
questBody

$requestHeader = @{ Authorization = "Bearer $($tokenResponse.access_token)
" }

# Get the business object summary for UserInfo. This will give us the busO
bId
$summaryUri = $baseUrl + "api/v1/getbusinessobjectsummary/busobname/UserIn
fo"
$summaryResults = Invoke-RestMethod -Method GET -Uri $summaryUri -ContentT
ype application/json
```



```
-Header $requestHeader
$busObjId = $summaryResults[0].busobjId

# Get the security group information
$securityGroupUri = $baseUri + "api/V2/getsecuritygroups"
$securityGroupResponse = Invoke-RestMethod -Method GET -Uri $securityGroupUri
Uri
-ContentType application/json -Header $requestHeader
$adminGroup = $securityGroupResponse.securityGroups | Where-Object {$_.groupName -eq "Admin"}

# Get the template for UserInfo. Use this to get field ids
$templateUri = $baseUri + "api/V1/getbusinessobjecttemplate"
$templateRequestBody =
@{
    busObjId = $busObjId
    includeAll = $true
} | ConvertTo-Json
$templateResponse = Invoke-RestMethod -Method POST -Uri $templateUri -ContentType application/json
-Header $requestHeader -Body $templateRequestBody

# Create the save request
$userSaveUri = $baseUri + "api/V2/saveuser"
$userSaveRequest =
@{
    AccountLocked = $false;
    BusObjId = $busObjId;
    DisplayName = "Test User";
    LdapRequired = $false;
    LoginId = "Test";
    NextPasswordResetDate = $null;
    Password = "Password";
    PasswordNeverExpires = $true;
```

```
SecurityGroupId = $adminGroup.groupId;
UserCannotChangePassword = $false;
UserMustChangePasswordAtNextLogin = $false;
UserInfoFields = @([
    @{
        Dirty = $true;
        Name = "FullName";
        Value = "Test User";
        FieldId = ($templateResponse.fields | Where-Object {$_.name
-eq "FullName"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "EmployeeID";
        Value = "123456";
        FieldId = ($templateResponse.fields | Where-Object
{$_.name -eq "EmployeeID"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "Comments";
        Value = "Created by API in PowerShell";
        FieldId = ($templateResponse.fields | Where-Object {$_.name
-eq "Comments"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "Department";
        Value = "IT";
        FieldId = ($templateResponse.fields | Where-Object
{$_.name -eq "Department"}).fieldId
    };
]);
```

```

    @{
        Dirty = $true;
        Name = "Office";
        Value = "Colorado Springs";
        FieldId = ($templateResponse.fields | Where-Object {$_.name
-eq "Office"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "Phone";
        Value = "719-777-7777";
        FieldId = ($templateResponse.fields | Where-Object {$_.name
-eq "Phone"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "CellPhone";
        Value = "719-777-7778";
        FieldId = ($templateResponse.fields | Where-Object
{$_.name -eq "CellPhone"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "FirstName";
        Value = "Test";
        FieldId = ($templateResponse.fields | Where-Object
{$_.name -eq "FirstName"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "MiddleInitial";
        Value = "C";
    }

```

```

        FieldId = ($templateResponse.fields | Where-Object
{$_ .name -eq "MiddleInitial"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "LastName";
        Value = "User";
        FieldId = ($templateResponse.fields | Where-Object {$_ .name
-eq "LastName"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "HomePhone";
        Value = "719-777-7779";
        FieldId = ($templateResponse.fields | Where-Object
{$_ .name -eq "HomePhone"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "Address";
        Value = "1234 Cherwell Ave";
        FieldId = ($templateResponse.fields | Where-Object {$_ .name
-eq "Address"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "City";
        Value = "Colorado Springs";
        FieldId = ($templateResponse.fields | Where-Object {$_ .name
-eq "City"}).fieldId
    };
    @{

```

```
        Dirty = $true;
        Name = "ProvinceStateName";
        Value = "CO";
        FieldId = ($templateResponse.fields | Where-Object

{$_ .name -eq "ProvinceStateName"}).fieldId
    };
    @{
        Dirty = $true;
        Name = "PostalCodeZip";
        Value = "80132";
        FieldId = ($templateResponse.fields | Where-Object

{$_ .name -eq "PostalCodeZip"}).fieldId
    }
)
} | ConvertTo-Json

# Create the user
$userSaveResponse = Invoke-RestMethod -Method POST -Uri $userSaveUri -Header $requestHeader
    -ContentType application/json -Body $userSaveRequest
```

# Operation Versioning

Cherwell REST API operation versioning is possible with these operations:

1. Add the operation version in the URL. (Example: `http://localhost/CherwellRESTAPI/api/v1/serviceinfo`)
2. Add the operation version in a custom header attribute. add `x-api-version` header to the request specifying the operation version number. (Example: `http://localhost/CherwellRESTAPI/api/serviceinfo` with a header `x-api-version: 1`)
3. Add a value to the Accept header. (Example: `http://localhost/CherwellRESTAPI/api/serviceinfo` with the Accept header `application/json ; api-version=1`)
4. Add a query string parameter to the URL. (Example: `http://localhost/CherwellRESTAPI/api/serviceinfo?v=1`)

The operation version of the Cherwell REST API differs from the underlying CSM version, though an installation is dependent on a version of CSM.

Operations will increment versions when request/response contracts are changed by removing properties, changing property data types, or changing the schema of complex data types of properties.

An operation version of an operation remains the same as long as there are no breaking changes to the request/response contracts.

The operation version will not be incremented for each CSM release unless breaking changes were made. New operations will be added as version 1 of those operations. If no version is specified, v1 is used. If a version is in the URL and that does not exist, the operation will not be found. If the version is in the URL, it will override all the other version specifying operations. For example, if operation 2, 3, or 4 are used, the URL cannot have a version segment.

## Errors

The Cherwell REST API uses conventional HTTP response codes to indicate the success or failure of an API request.

In general:

- Codes in the 200 range indicate success.
- Codes in the 400 range indicate a failure error based on the information provided.

Here are a few http codes that you might see:

200	OK. Everything worked as expected.
400	Bad request, often due to a missing required parameter.
402	Request failed. Typically, the parameters are valid but the request failed.
404	Not found. The requested resource does not exist.

Not all errors map cleanly to HTTP response codes. In these cases, check the error message for more information.

For example, if you specify an incorrect client ID, you will get the following response:

```
Status Code 400
{"error":"invalid_client_id","error_description":"Client 12345 is not registered in the system.\"}"
```

If you supply invalid credentials, you will get the following response:

```
Status Code 400
{"error":"invalid_grant","error_description":"Could not login to system with the provided user ID and password.\"}"
```

# Using Swagger Code Generation

Use swagger-codegen to generate client code.



**Important:** Swagger categories were re-organized for CSM version 9.2.0. Your project will return errors if you re-generate Swagger code in order to access new API operations after upgrading to version 9.2.0 or later. For details, refer to the section [Upgrade Considerations](#) in Upgrade Considerations.

To get started, refer to the [swagger-codegen GitHub](#).

Follow this process:

1. Install [Apache Maven](#).
2. Download the swagger-codegen source code from the GitHub repository.
3. From a command prompt at the root directory of the Swagger source code, run the command mvn package.
4. Create a .bat file similar to the following example:

```
java -jar modules/swagger-codegen-cli/target/swagger-codegen-cli.jar generate -i http://localhost/CherwellApi/swagger/docs/all -l csharp -o cherwell/client/csharp
```

5. Place the .bat file in the same level as the Swagger code gen modules directory.



**Note:** The example script creates output in a cherwell/client/csharp directory at the same level where the .bat file is placed.

6. Add the generated code from the output directory to your project and reference it.

You can now call the generated Swagger code for CSM.

## Sample Authenticated Request

```
var service = new ServiceApi(baseuri);
var tokenResponse = service.ServiceToken("password", clientid, "", username, password, "", "Internal");
var searchesApi = new SearchesApi(Baseuri);
searchesApi.DefaultHeader.Add("Authorization", "Bearer " + tokenResponse.AccessToken);
var searchItemResponse = searchesApi.SearchesGetSearchItemsV1(null);
```





# Operations List

Cherwell REST API operations are organized by category and ordered by URL.

You can explore operations using the [Rest API Discovery Tool](#).

# Business Object Operations

Operations are ordered by URL.

## Delete Business Objects in a Batch

- **URL**  
POST /api/v1/deletebusinessobjectbatch/
- **Description:**  
Delete Business Objects in a batch using either the public ID or record ID. Batch can contain multiple Business Object types.

## Delete Business Object by Public ID

- **URL**  
DELETE /api/v1/deletebusinessobject/busobid/{busobid}/publicid/{publicid}
- **Description**  
Delete a single Business Object using its Public ID.

## Delete Business Object by Record ID

- **URL**  
DELETE /api/v1/deletebusinessobject/busobid/{busobid}/busobrecid/{busobrecid}
- **Description**  
Delete a single Business Object using its record ID.

## Delete Related Business Object by Public ID

- **URL**  
DELETE /api/v1/deleterelatedbusinessobject/parentbusobid/{parentbusobid}/parentbusobrecid/{parentbusobrecid}/relationshipid/{relationshipid}/publicid/{publicid}
- **Description**  
Delete a related Business Object using its public ID.  
  
Use "Unlink Related Business Object" to unlink two Business Objects rather than deleting the related Business Object.

## Delete Related Business Object by Record ID

- **URL**

```
DELETE /api/v1/deleterelatedbusinessobject/parentbusobid/  
{parentbusobid}/parentbusobrecid/{parentbusobrecid}/relationshipid/  
{relationshipid}/busobrecid/{busobrecid}
```

- **Description**

Delete a related Business Object using its record ID.

Use "Unlink Related Business Object" to unlink two Business Objects rather than deleting the related Business Object.

## Get Lookup Values for Fields

- **URL**

```
POST /api/v1/fieldvalueslookup
```

- **Description**

Get potentially valid values for Business Object fields.

## Get Activities

- **URL**

```
GET /api/v1/getactivities/busobid/{busobid}/busobrecid/{busobrecid}/  
pagesize/{pagesize}
```

- **Description**

Get mapped Activity types for Audit, Communication, and Notes.

## Get an Imported Business Object Attachment

- **URL**

```
GET /api/v1/getbusinessobjectattachment/attachmentid/{attachmentid}/  
busobid/{busobid}/busobrecid/{busobrecid}
```

- **Description**

Get a Business Object attachment that has been imported into the system. HTTP Range Header can be used but is optional.

## Get Attachments by Business Object Public ID

- **URL**

```
GET /api/v1/getbusinessobjectattachments/busobid/{busobid}/publicid/  
{publicid}/type/{type}/attachmenttype/{attachmenttype}
```

- **Description**

Get attachments for a Business Object by Business Object ID and public ID.

## Get Attachments by Business Object Record ID

- **URL**  
GET /api/V1/getbusinessobjectattachments/busobid/{busobid}/busobrecid/{busobrecid}/type/{type}/attachmenttype/{attachmenttype}
- **Description**  
Get attachments for a Business Object by Business Object ID and record ID.

## Get Attachments by Business Object Name and Public ID

- **URL**  
GET /api/V1/getbusinessobjectattachments/busobname/{busobname}/publicid/{publicid}/type/{type}/attachmenttype/{attachmenttype}
- **Description**  
Get attachments for a Business Object by Business Object name and record ID.

## Get Attachments by Business Object Name and Record ID

- **URL**  
GET /api/V1/getbusinessobjectattachments/busobname/{busobname}/busobrecid/{busobrecid}/type/{type}/attachmenttype/{attachmenttype}
- **Description**  
Get attachments for a Business Object by Business Object name and record ID.

## Get Business Object Attachments by Request Object

- **URL**  
POST /api/V1/getbusinessobjectattachments
- **Description**  
Get attachments for a Business Object by attachments request object. You can also request a list of types to get more than one type at a time.

## Get a Batch of Business Objects

- **URL**  
POST /api/V1/getbusinessobjectbatch
- **Description**  
Get a batch of Business Object records that includes a list of field record IDs, display names, and values for each record.

### Get a Business Object Record by Public ID

- **URL**  
GET /api/V1/getbusinessobject/busobid/{busobid}/publicid/{publicid}
- **Description**  
Get a Business Object record that includes a list of fields and their record IDs, names, and set values.

### Get a Business Object Record by Record ID

- **URL**  
GET /api/V1/getbusinessobject/busobid/{busobid}/busobrecid/{busobrecid}
- **Description**  
Get a Business Object record that includes a list of fields and their record IDs, names, and set values.

### Get a Business Object by Scan Code and Business Object ID

- **URL**  
GET /api/V1/getbusinessobject/scancode/{scanCode}/busobid/{busobid}
- **Description**  
Get a Business Object record based on its associated scan code and Business Object ID.

### Get a Business Object by Scan Code and Business Object Name

- **URL**  
GET /api/V1/getbusinessobject/scancode/{scanCode}/busobname/{busobname}
- **Description**  
Get a Business Object record based on its associated scan code and Business Object name.

### Get a Business Object Schema

- **URL**  
GET /api/V1/getbusinessobjectschema/busobid/{busobId}
- **Description**  
Get the schema for a Business Object and, optionally, its related Business Objects.

### Get Business Object Summaries by Type

- **URL**  
GET /api/V1/getbusinessobjectsummaries/type/{type}

- **Description**  
Get a list of Business Object summaries by type (Major, Supporting, Lookup, Groups, and All).

### Get a Business Object Summary by ID

- **URL**  
GET /api/V1/getbusinessobjectsummary/busobid/{busobid}
- **Description**  
Get a single Business Object summary by ID.

### Get a Business Object Summary by Name

- **URL**  
GET /api/V1/getbusinessobjectsummary/busobname/{busobname}
- **Description**  
Get a single Business Object summary by name.

### Get a Business Object Template

- **URL**  
POST /api/V1/getbusinessobjecttemplate
- **Description**  
Get a template used to create Business Objects. The template includes placeholders for field values. You can then send the template with these values to the Business Object Save operation.



**Important:** Use only HTML in the HTML field value so that the field is read as expected. For example, use the HTML <br> tag to produce a line break instead of the Windows \r\n.

### Get Related Business Objects by Request Object

- **URL**  
POST /api/V1/getrelatedbusinessobject
- **Description**  
Get related Business Objects for a specific Relationship. Specify a list of fields to include in the response.

### Get Related Business Objects by ID

- **URL**  
GET /api/V1/getrelatedbusinessobject/parentbusobid/{parentbusobid}/parentbusobrecid/{parentbusobrecid}/relationshipid/{relationshipid}

- **Description**  
Get the related objects for a Business Object relationship specifying all fields or default grid as the field to return.

### Get Related Business Objects (Custom Grid)

- **URL**  
GET /api/V1/getrelatedbusinessobject/parentbusobid/{parentbusobid}/parentbusobrecid/{parentbusobrecid}/relationshipid/{relationshipid}/gridid/{gridid}
- **Description**  
Get related Business Objects for a specific relationship. Specify a custom grid ID as the fields to return.

### Link Related Business Objects

- **URL**  
GET /api/V1/linkrelatedbusinessobject/parentbusobid/{parentbusobid}/parentbusobrecid/{parentbusobrecid}/relationshipid/{relationshipid}/busobid/{busobid}/busobrecid/{busobrecid}
- **Description**  
Link related Business Objects.

### Remove an Attachment by Business Object ID and Public ID

- **URL**  
DELETE /api/V1/removebusinessobjectattachment/attachmentid/{attachmentid}/busobid/{busobid}/publicid/{publicid}
- **Description**  
Remove an attachment from a Business Object using the attachment record ID, Business Object ID, and Business Object public ID.

### Remove an Attachment by Business Object ID and Record ID

- **URL**  
DELETE /api/V1/removebusinessobjectattachment/attachmentid/{attachmentid}/busobid/{busobid}/busobrecid/{busobrecid}
- **Description**  
Remove an attachment from a Business Object using the attachment record ID, Business Object ID, and Business Object record ID.



## Remove an Attachment by Business Object Name and Public ID

- **URL**  
DELETE /api/V1/removebusinessobjectattachment/attachmentid/{attachmentid}/busobname/{busobname}/publicid/{publicid}
- **Description**  
Remove an attachment from a Business Object using the attachment record ID, Business Object name, and Business Object record ID.

## Remove an Attachment by Business Object Name and Record ID

- **URL**  
DELETE /api/V1/removebusinessobjectattachment/attachmentid/{attachmentid}/busobname/{busobname}/busobrecid/{busobrecid}
- **Description**  
Remove an attachment from a Business Object using the attachment record ID, Business Object name, and Business Object public ID.

## Attach a Business Object to Another Business Object

- **URL**  
PUT /api/V1/savebusinessobjectattachmentbusob
- **Description**  
Attach a Business Object to a Business Object. This links the Business Object but does not create a relationship between the two. (Use "Link Related Business Objects" to create a relationship.)

## Attach a File via UNC

- **URL**  
PUT /api/V1/savebusinessobjectattachmenturl
- **Description**  
Attach a URL path to a Business Object.

## Attach a URL Path

- **URL**  
PUT /api/V1/savebusinessobjectattachmentlink
- **Description**  
Attach a file to a Business Object via a path (UNC recommended).

## Create or Update a Batch of Business Objects

- **URL**  
POST /api/V1/savebusinessobjectbatch
- **Description**  
Create or update an array of Business Objects in a batch. To update, specify record ID or public ID. To create, leave record ID and public ID empty.

## Create or Update a Business Object

- **URL**  
POST /api/V1/savebusinessobject
- **Description**  
Create a new Business Object or update an existing Business Object. To update, specify record ID or public ID. To create, leave record ID and public ID empty.

## Create or Update a Related Business Object

- **URL**  
POST /api/V1/saverelatedbusinessobject
- **Description**  
Create or update a related Business Object. To update, specify record ID or public ID. To create, leave record ID and public ID empty.

## Unlink Related Business Objects

- **URL**  
DELETE /api/V1/unlinkrelatedbusinessobject/parentbusobid/  
{parentbusobid}/parentbusobrecid/{parentbusobrecid}/relationshipid/  
{relationshipid}/busobid/{busobid}/busobrecid/{busobrecid}
- **Description**  
Unlink related Business Objects.

## Upload an Attachment by Business Object ID and Public ID

- **URL**  
POST /api/V1/uploadbusinessobjectattachment/filename/{filename}/busobid/  
{busobid}/publicid/{publicid}/offset/{offset}/totalsize/{totalsize}
- **Description**  
Upload an attachment to a Business Object record using a Business Object ID and public ID.

### Upload an Attachment by Business Object ID and Record ID

- **URL**  
POST /api/V1/uploadbusinessobjectattachment/filename/{filename}/busobid/{busobid}/busobrecid/{busobrecid}/offset/{offset}/totalsize/{totalsize}
- **Description**  
Upload an attachment to a Business Object record using a Business Object ID and record ID.

### Upload an Attachment by Business Object Name and Public ID

- **URL**  
POST /api/V1/uploadbusinessobjectattachment/filename/{filename}/busobname/{busobname}/publicid/{publicid}/offset/{offset}/totalsize/{totalsize}
- **Description**  
Upload an attachment to a Business Object record using a Business Object name and public ID.

### Upload an Attachment by Business Object Name and Record ID

- **URL**  
POST /api/V1/uploadbusinessobjectattachment/filename/{filename}/busobname/{busobname}/busobrecid/{busobrecid}/offset/{offset}/totalsize/{totalsize}
- **Description**  
Upload an attachment to a Business Object record using a Business Object name and record ID.

# Canonical Operations

Refer to [About the Cherwell Canonical REST API](#) for information about the Cherwell Canonical REST API.

These operations are primarily intended for customers and vendors to create reusable integrations between third-party software and customized CSM systems. If you are a Cherwell Administrator and need to integrate with your instance of CSM, we recommend beginning with the Named Object Operations.



**Note:** For the operation URLs listed below, `{BusObjName}` is replaced in Swagger with the canonical name of the Business Object on which the action is performed, and `{Version}` is replaced with the version of the canonical Business Object Schema. For example, for a custom Business Object called Ticket that is mapped to version one of the canonical Schema for Incident, the Describe Business Object URL is `GET /api/V1/canonical/Incident`.

Operations are ordered by URL.

## Describe Business Object

- **URL**  
`GET /api/{Version}/canonical/{BusObjName}`
- **Description:**  
Get the Business Object Schema using the canonical Business Object name.

## Create Business Object

- **URL**  
`POST /api/{Version}/canonical/{BusObjName}`
- **Description:**  
Create a new Business Object.

## Delete Business Object by Business Object ID

- **URL**  
`DELETE /api/{Version}/canonical/{BusObjName}/{ID}`
- **Description:**  
Delete a Business Object using the Business Object ID.

## Update Business Object by Business Object ID

- **URL**  
`PATCH /api/{Version}/canonical/{BusObjName}/{ID}`

- **Description:**  
Update a Business Object using the Business Object ID.

## Search for Business Object by Search Term, Value, or Date Range

- **URL**  
GET /api/{Version}/canonical/{BusObName}/search
- **Description:**  
Search for a Business Object using a search term, value, or date range. You can specify one or more search criteria. The search parameters are appended to the URL.

## Search for Business Object

- **URL**  
POST /api/{Version}/canonical/{BusObName}/search
- **Description:**  
Search for a Business Object. You can specify one or more search criteria. The search parameters are contained within the request body.

## Get Business Object Comments

- **URL**  
GET /api/{Version}/canonical/{BusObName}/{id}/Comments
- **Description:**  
Get comments associated with a Business Object using the Business Object ID.

## Create a Business Object Comment

- **URL**  
POST /api/{Version}/canonical/{BusObName}/{id}/Comments
- **Description:**  
Create a comment for a Business Object using the Business Object ID.

## Delete a Business Object Comment

- **URL**  
POST /api/{Version}/canonical/{BusObName}/Comments/{id}
- **Description:**  
Delete a comment for a Business Object using the comment ID.

## Describe Business Object Comments

- **URL**  
GET /api/{Version}/canonical/{BusObName}/Comments
- **Description:**  
Get the Schema for the Business Object comments.

# Core Operations

## Delete a Gallery Image

- **URL**  
DELETE /api/V1/deletegalleryimage/standinkey/{standinkey}
- **Description**  
Operation to delete a gallery image using the stand in key.

## Get Gallery Images by Scope, Scope Owner, and Folder

- **URL**  
GET /api/V1/getgalleryimages/scope/{scope}/scopeowner/{scopeowner}/folder/{folder}
- **Description**  
Operation to get gallery images for the specified Scope, Scope Owner, and Folder.

## Get Gallery Images by Scope and Scope Owner

- **URL**  
GET /api/V1/getgalleryimages/scope/{scope}/scopeowner/{scopeowner}
- **Description**  
Operation to get gallery images for the specified Scope and Scope Owner.

## Get Gallery Images by Scope

- **URL**  
GET /api/V1/getgalleryimages/scope/{scope}
- **Description**  
Operation to get all gallery images for the specified Scope.

## Get All Gallery Images

- **URL**  
GET /api/V1/getgalleryimages
- **Description**  
Operation to get all gallery images in the system.

## Get Built-in Images

- **URL**  
GET /api/V1/getgalleryimage/name/{name}

- **Description**  
Operation to get built-in images. If you are requesting an icon (.ico), you can specify width and height.

### Get Stored Values by Folder

- **URL**  
GET /api/V1/storedvalues/scope/{scope}/scopeowner/{scopeowner}/folder/{folder}
- **Description**  
Operation to get Stored Values for the specified Folder.

### Get Stored Values by Scope and Scope Owner

- **URL**  
GET /api/V1/storedvalues/scope/{scope}/scopeowner/{scopeowner}
- **Description**  
Operation to get Stored Values for the specified Scope and Scope Owner.

### Get Stored Values by Scope

- **URL**  
GET /api/V1/storedvalues/scope/{scope}
- **Description**  
Operation to get Stored Values for the specified Scope.

### Get All Stored Values in the System

- **URL**  
GET /api/V1/storedvalues
- **Description**  
Operation to get all Stored Values in the system.

### Get a Stored Value by Stand In Key

- **URL**  
GET /api/V1/getstoredvalue/standinkey/{standInKey}
- **Description**  
Operation to get a Stored Value by stand in key.



## Get a List of Views

- **URL**  
GET /api/V1/getviews
- **Description**  
Operation to get a list of all views that are configured in the system.

## Create or Update a Gallery Image

- **URL**  
POST /api/V1/savegalleryimage
- **Description**  
Operation to create or update a gallery image. To create a new gallery image, do not include a stand in key. To update a gallery image, provide the stand in key for the image.  
  
Allowable image types include: imported, URL, and file. To use the imported image type, use the name property to specify the file name with extension and the Base64EncodedImageData property to specify the image data in base 64 encoded format. The maximum file size is 512KB.  
  
To use the URL image type, use the name property to specify the full network share path to the file, for example "\\networkshare\\somefolder\\somefile.jpg". If the file is not accessible to all users it will not be visible to all users.  
  
To use the file image type, use the name property to specify the full path to the file, for example "C:\\somefolder\\somfile.jpg". If the file is not accessible to all users it will not be visible to all users.  
  
When creating or updating an image, name and image type are always required. If the image type is "Imported" the base 64 encoded image data is also required.  
  
Scope, Scope Owner, and Folder can all be updated independently.

## Create or Update a Stored Value

- **URL**  
POST /api/V1/savestoredvalue
- **Description**  
Operation to create or update a Stored Value. To update, specify the stand in key for the Stored Value to update. To create, leave the stand in key property blank, and provide a name, scope, type, and value.

## Set the Culture for the Current User

- **URL**  
PUT /api/V1/setculture/culturecode/{culturecode}
- **Description**  
Operation to update the culture by culture code for the current User. This returns a new access token that contains the updated information.



# Form Operations

Operations are ordered by URL.

## Get Mobile Form by Business Object ID and Public ID

- **URL**  
GET /api/V1/getmobileformforbusob/busobid/{busobid}/publicid/{publicid}
- **Description**  
Get a Mobile Form for a specific business object by Business Object ID and Public ID.

## Get Mobile Form by Business Object ID and Record ID

- **URL**  
GET /api/V1/getmobileformforbusob/busobid/{busobid}/busobrecid/{busobrecid}
- **Description**  
Get a Mobile Form for a specific business object by Business Object ID and record ID.

## Get Mobile Form by Business Object Name and Public ID

- **URL**  
GET /api/V1/getmobileformforbusob/busobname/{busobname}/publicid/{publicid}
- **Description**  
Get a Mobile Form for a specific business object by Business Object name and public ID.

## Get Mobile Form by Business Object Name and Record ID

- **URL**  
GET /api/V1/getmobileformforbusob/busobname/{busobname}/busobrecid/{busobrecid}
- **Description**  
Get a Mobile Form for a specific business object by Business Object name and record ID.

# Named Object Operations

Use Named Object Operations to perform basic actions against Business Objects. Named Object Operations include nine API operations with simplified URL paths that you can run against any major Business Object.

Named Object Operations are not automatically visible in Swagger. To add the Named Object Operations for a specific Business Object, follow the steps to [Add Operations for Named Objects](#).



**Important:** For the operation URLs listed below, {BusObjectName} is replaced in Swagger with the internal name of the Business Object on which the action is performed. For example, for Incident, the Delete Business Object URL is DELETE /api/v1/object/Incident/{id}.

Operations are ordered by URL.

## Delete Business Object

- **URL**  
DELETE /api/v1/object/{BusObjectName}/{id}
- **Description:**  
Delete a Business Object using its name and ID.

## Get Business Object

- **URL**  
GET /api/v1/object/{BusObjectName}/{id}
- **Description:**  
Get a Business Object using its name and ID.

## Update Business Object

- **URL**  
PATCH /api/v1/object/{BusObjectName}/{id}
- **Description:**  
Update a Business Object using its name, ID, and properties.

## Create Business Object

- **URL**  
POST /api/v1/object/{BusObjectName}
- **Description:**  
Create a Business Object using its name and properties.

### Get Business Object Field List

- **URL**  
GET /api/V1/object/{BusObName}/describe
- **Description:**  
Get a Business Object Field list using its name.

### Delete Business Object Comment

- **URL**  
DELETE /api/V1/object/{BusObName}/{id}/comments/{commentid}
- **Description:**  
Delete a Business Object comment using its name, ID, and comment ID.

### Get Business Object Comments

- **URL**  
GET /api/V1/object/{BusObName}/{id}/comments
- **Description:**  
Get a list of comments for a Business Object using its name and ID.

### Create Business Object Comment

- **URL**  
POST /api/V1/object/{BusObName}/{id}/comments
- **Description:**  
Create a comment for a Business Object using its name and ID.

### Search for Business Object

- **URL**  
POST /api/V1/object/{BusObName}/search
- **Description:**  
Search for a Business Object using its name and an array of filters.

## Add Operations for Named Objects

Add the PublishToApi attribute to a Business Object so that its REST API operations become available in Named Object Operations.

1. Open CSM Administrator, and create a new Blueprint.
2. Select a Business Object from the Object Manager.
3. Click **Edit Business Object** under Business Object tasks.  
The **Edit Business Object** page appears.
4. Click the **Bus Ob Properties** button.  
The **Business Object Properties** dialog opens.
5. Click **Advanced** and expand **General Attributes**.
6. In the **Attribute** column, type **PublishToApi**. Leave the corresponding **Value** field empty.
7. Click **OK** and save the Blueprint.
8. Repeat steps 2 through 7 for additional Business Objects, if necessary.
9. Publish the Blueprint.  
REST API operations for the selected Business Objects are available in the Named Object Operations section of Swagger.

# Queue Operations

## Add a Business Object to a Queue

- **URL**  
POST /api/V1/additemtoqueue
- **Description**  
Operation to add a Business Object to a Queue. To get the Business Object ID, use "Get Business Object Summary by Name."

## Check a Business Object in to a Queue

- **URL**  
POST /api/V1/checkinqueueitem
- **Description**  
Operation to check a Business Object in to a Queue. To get the Business Object ID, use "Get Business Object Summary by Name."

## Check a Business Object out of a Queue

- **URL**  
POST /api/V1/checkinqueueitem
- **Description**  
Operation to check a Business Object out of a Queue. To get the Business Object ID, use "Get Business Object Summary by Name." To get the Queue stand in key use "Get Available Queues."

## Get Available Queues by Scope, Scope Owner, and Folder

- **URL**  
GET /api/V1/getqueues/scope/{scope}/scopeowner/{scopeowner}/folder/{folder}
- **Description**  
Operation to get available Queues for a specific Business Object type by Scope, Scope Owner, and Folder.

## Get Available Queues by Scope and Scope Owner

- **URL**  
GET /api/V1/getqueues/scope/{scope}/scopeowner/{scopeowner}
- **Description**  
Operation to get available Queues for a specific Business Object type by Scope and Scope Owner.

### Get Available Queues by Scope

- **URL**  
GET /api/V1/getqueues/scope/{scope}
- **Description**  
Operation to get available Queues for a specific Business Object type by Scope.

### Get All Available Queues

- **URL**  
GET /api/V1/getqueues
- **Description**  
Operation to get available Queues for a specific Business Object.

### Remove a Business Object From a Queue

- **URL**  
POST /api/V1/removeitemfromqueue
- **Description**  
Operation to remove an item from a Queue. To get the Business Object ID, use "Get Business Object Summary by Name." To get the Queue stand in key use "Get Available Queues."



# Search Operations

Operations are ordered by URL.

## Get a Quick Search Configuration for a List of Business Objects

- **URL**  
POST /api/V1/getquicksearchconfigurationforbusobs
- **Description**  
Build a Quick Search configuration that you can use to execute a Quick Search for multiple Business Objects.

## Get a Quick Search Configuration with View Rights

- **URL**  
GET /api/V1/getquicksearchconfigurationforbusobswithviewrights
- **Description**  
Get a Quick Search configuration that you can use to execute a Quick Search based the current User's Business Object view rights.

## Get Quick Search Results for a List of Business Objects

- **URL**  
POST /api/V1/getquicksearchresults
- **Description**  
Execute a Quick Search using a list of Business Object IDs and search text.

## Get Quick Search Results for a Specific Business Object

- **URL**  
POST /api/V1/getquicksearchspecificresults  
POST /api/V2/getquicksearchspecificresults
- **Description**  
Execute a Quick Search for a specific Business Object.

## Get All Saved Searches by Folder ID

- **URL**  
GET /api/V1/getsearchitems/association/{association}/scope/{scope}/scopeowner/{scopeowner}/folder/{folder}  
GET /api/V2/getsearchitems/association/{association}/scope/{scope}/scopeowner/{scopeowner}/folder/{folder}

- **Description**

Get a tree of saved queries using Business Object Association, Scope, Scope Owner, and Folder. Tree includes Scope, search name, IDs, and location within the tree.

### Get All Saved Searches by Scope Owner

- **URL**

GET /api/V1/getsearchitems/association/{association}/scope/{scope}/scopeowner/{scopeowner}

GET /api/V2/getsearchitems/association/{association}/scope/{scope}/scopeowner/{scopeowner}

- **Description**

Get a tree of saved queries using Business Object Association, Scope, and Scope Owner. Tree includes Scope, search name, IDs, and location within the tree.

### Get All Saved Searches by Scope

- **URL**

GET /api/V1/getsearchitems/association/{association}/scope/{scope}

GET /api/V2/getsearchitems/association/{association}/scope/{scope}

- **Description**

Get a tree of saved queries using Business Object Association and Scope. Tree includes Scope, search name, IDs, and location within the tree.

### Get All Saved Searches by Business Object Association

- **URL**

GET /api/V1/getsearchitems/association/{association}

GET /api/V2/getsearchitems/association/{association}

- **Description**

Get a tree of saved queries using Business Object Association. Tree includes Scope, search name, IDs, and location within the tree.

### Get All Saved Searches by Default Business Object Association

- **URL**

GET /api/V1/getsearchitems

GET /api/V2/getsearchitems

- **Description**

Get a tree of saved queries using the default Business Object Association. Tree includes Scope, search name, IDs, and location within the tree.

## Run an Ad-Hoc Search

- **URL**  
POST /api/V1/getsearchresults
- **Description**  
Run an ad-hoc Business Object search. To execute a search with Prompts, the PromptId and Value are required in the Prompt request object.

## Run a Saved Search by ID

- **URL**  
GET /api/V1/getsearchresults/association/{association}/scope/{scope}/scopeowner/{scopeowner}/searchid/{searchid}
- **Description**  
Get the paged results of a saved search by Business Object association, Scope, Scope Owner, and Search ID.  
  
When the search contains Prompts, the response contains the Prompt. Send the Prompt and the original operation parameters to SearchResultsRequest to the getsearchresults ad-hoc http post operation.

## Run a Saved Search by Name

- **URL**  
GET /api/V1/getsearchresults/association/{association}/scope/{scope}/scopeowner/{scopeowner}/searchname/{searchname}
- **Description**  
Get the paged results of a saved search by Business Object association, Scope, Scope Owner, and Search Name.  
  
When the search contains Prompts, the response contains the Prompt. Send the Prompt and the original operation parameters to SearchResultsRequest to the getsearchresults ad-hoc http post operation.

## Export an Ad-Hoc Search

- **URL**  
POST /api/V1/getsearchresultsexport
- **Description**  
Operation that returns an ad-hoc search in a specified export format: 0=CSV, 1=Excel, 2=Tab, 3=Word, 4=Custom Separator, 5=Simple JSON.  
  
To execute a search with Prompts, the PromptId and Value are required in the Prompt request object.

## Export a Saved Search by ID

- **URL**

GET /api/V1/getsearchresultsexport/association/{association}/scope/{scope}/scopeowner/{scopeowner}/searchid/{searchid}/exportformat/{exportformat}

- **Description**

Operation that returns a saved search using Business Object Association, Scope, Scope Owner, Search ID, and export format.

When the search contains Prompts, the response contains the Prompt. Send the Prompt and the original operation parameters to SearchResultsRequest to the getsearchresultsexport ad-hoc http post operation.

## Export a Saved Search by Name

- **URL**

GET /api/V1/getsearchresultsexport/association/{association}/scope/{scope}/scopeowner/{scopeowner}/searchname/{searchname}/exportformat/{exportformat}

- **Description**

Operation that returns a saved search using Business Object Association, Scope, Scope Owner, Search Name, and export format.

When the search contains Prompts, the response contains the Prompt. Send the Prompt and the original operation parameters to SearchResultsRequest to the getsearchresultsexport ad-hoc http post operation.

# Security Operations

Operations are ordered by URL.

## Get Client Security Settings

### URL

GET /api/V1/getclientsecuritysettings/applicationtype/{applicationtype}

### Description

Get configured security settings for specified CSM clients.

## Get Roles

- **URL**

GET /api/V1/getroles

GET /api/V2/getroles

- **Description**

Get all available roles.

## Get Security Group Permissions by Business Object ID

- **URL**

GET /api/V1/getsecuritygroupbusinessobjectpermissions/groupid/{groupid}/busobid/{busObId}

GET /api/V2/getsecuritygroupbusinessobjectpermissions/groupid/{groupid}/busobid/{busObId}

- **Description**

Get specific Business Object permissions for a Security Group.

## Get Security Group Permissions by Business Object Name

- **URL**

GET /api/V1/getsecuritygroupbusinessobjectpermissions/groupid/{groupid}/busobname/{busobname}

GET /api/V2/getsecuritygroupbusinessobjectpermissions/groupid/{groupid}/busobname/{busobname}

- **Description**

Get specific Business Object permissions for a Security Group.

## Get Current User's Permissions by Business Object ID

- **URL**

```
GET /api/V1/  
getsecuritygroupbusinessobjectpermissionsforcurrentuserbybusobid/  
busobid/{busObId}
```

```
GET /api/V2/  
getsecuritygroupbusinessobjectpermissionsforcurrentuserbybusobid/  
busobid/{busObId}
```

- **Description**

Get Business Object permissions for the currently logged-in User's Security Group.

## Get Current User's Permissions by Business Object Name

- **URL**

```
GET /api/V1/  
getsecuritygroupbusinessobjectpermissionsforcurrentuserbybusobname/  
busobname/{busobname}
```

```
GET /api/V2/  
getsecuritygroupbusinessobjectpermissionsforcurrentuserbybusobname/  
busobname/{busobname}
```

- **Description**

Get Business Object permissions for the currently logged-in User's Security Group.

## Get Security Group Categories

- **URL**

```
GET /api/V1/getsecuritygroupcategories
```

```
GET /api/V2/getsecuritygroupcategories
```

- **Description**

Get IDs and names for all available Security Group categories.

## Get Security Group Rights by Group and Category ID

- **URL**

```
GET /api/V1/getsecuritygrouprights/groupid/{groupid}/categoryid/  
{categoryid}
```

```
GET /api/V2/getsecuritygrouprights/groupid/{groupid}/categoryid/  
{categoryid}
```

- **Description**

Get rights for a Security Group by group ID and category ID.

## Get Security Group Rights by Group and Category Name

- **URL**

```
GET /api/V1/getsecuritygrouprights/groupname/{groupname}/categoryname/  
{categoryname}
```

```
GET /api/V2/getsecuritygrouprights/groupname/{groupname}/categoryname/  
{categoryname}
```

- **Description**

Get rights for a Security Group by group name and category name.

## Get Rights for Current User by Security Group Category ID

- **URL**

```
GET /api/V1/getsecuritygrouprightsforcurrentuserbycategoryid/categoryid/  
{categoryid}
```

```
GET /api/V2/getsecuritygrouprightsforcurrentuserbycategoryid/categoryid/  
{categoryid}
```

- **Description**

Get rights for the current User's Security Group by category ID.

## Get Rights for Current User by Security Group Category Name

- **URL**

```
GET /api/V1/getsecuritygrouprightsforcurrentuserbycategoryname/  
categoryname/{categoryname}
```

```
GET /api/V2/getsecuritygrouprightsforcurrentuserbycategoryname/  
categoryname/{categoryname}
```

- **Description**

Get rights for the current User's Security Group by category name.

## Get Available Security Groups

- **URL**

```
GET /api/V1/getsecuritygroups
```

```
GET /api/V2/getsecuritygroups
```

- **Description**

Get IDs, names, and descriptions for all available Security Groups.

## Get Users in a Security Group

- **URL**

```
GET /api/V1/getusersinsecuritygroup/groupid/{groupid}
```

```
GET /api/V2/getusersinsecuritygroup/groupid/{groupid}
```

- **Description**

Get all Users in a specified Security Group.



# Service Operations

Operations are ordered by URL.

## Log Out User by Token

- **URL**  
DELETE /api/V1/logout
- **Description**  
Log out the user referenced in an authentication token.

## Get REST API and CSM Information

- **URL**  
GET /api/V1/serviceinfo
- **Description**  
Get the latest REST API operation version, CSM version, and CSM system date and time.

## Get an Access Token

- **URL**  
POST /api/V1/token
- **Description**  
Request an access token for one of the following authentication modes. Or, you can request an access token using a refresh token. An API client key is required in both cases, and the authentication mode you use must be the mode used by the CSM Browser Client.
  - **Internal** - Use a CSM username and password. If no other mode is specified, Internal mode is used.
  - **Windows** - Uses the server variable LOGON\_USER to attempt to find a CSM user. You can also use domain\username and password.
  - **LDAP** - Uses the LDAP settings configured for CSM and the server variable LOGON\_USER to attempt to find a CSM user. You can also use domain\username and password.
  - **SAML** - Uses the SAML settings configured for CSM to validate credentials and find the CSM user.

# Team Operations

Operations are ordered by URL.

## Add a Batch of Users to a Team

- **URL**  
POST /api/V1/addusertoteambybatch
- **Description**  
Add a user to a Team. To get internal IDs for Users, use "Get a User by login ID" or "Get a User by public ID." To get a Team's internal ID, use "Get all available Teams."

## Add a User to a Team

- **URL**  
POST /api/V1/addusertoteam  
POST /api/V2/addusertoteam
- **Description**  
Add a User to a Team. To get the User's internal ID, use "Get a User by login ID" or "Get a User by public ID." To get a Team's internal ID, use "Get all available Teams."

## Delete a Team by Team ID

- **URL**  
DELETE /api/V1/deleteteam/{teamid}
- **Description:**  
Delete a single Team using its Team ID.

## Get Available Teams

- **URL**  
GET /api/V1/getteams  
GET /api/V2/getteams
- **Description**  
Get IDs, names, and descriptions for all available Teams.

## Get a Team by Team ID

- **URL**  
GET /api/V1/getteam/{teamid}
- **Description:**  
Get Team Info for a single Team using its Team ID.

## Get Team Assignments for a User

- **URL**  
GET /api/V1/getusersteams/userrecordid/{userRecordId}  
GET /api/V2/getusersteams/userrecordid/{userRecordId}
- **Description**  
Get Team assignments for a User.

## Get Available Workgroups

- **URL**  
GET /api/V1/getworkgroups  
GET /api/V2/getworkgroups
- **Description**  
Get IDs and names for all available Workgroups.

## Remove a User from a Team

- **URL**  
DELETE /api/V1/removeuserfromteam/teamid/{teamId}/userrecordid/{userrecordid}  
DELETE /api/V2/removeuserfromteam/teamid/{teamId}/userrecordid/{userrecordid}
- **Description**  
Remove a User from a Team. To remove a User, specify the Team ID and the User record ID.

## Remove a Customer from a Workgroup

- **URL**  
/api/V1/removecustomerfromworkgroup/workgroupid/{workgroupid}/customerrecordid/{customerrecordid}
- **Description**  
Remove a Customer from a Workgroup.

## Create or Update a Team

- **URL**  
POST /api/V1/saveteam
- **Description**  
Create or update a Team. To update, specify Team ID and Team type. To create, specify Team name and Team type.

### **Add or Update a Team Member**

- **URL**  
POST /api/V1/saveteammember
- **Description**  
Add or update a Team Member. Specify User ID, Team ID, and if Team Manager. Optionally, set the Team as the default Team.

### **Add or Update a Workgroup Member**

- **URL**  
POST /api/V1/saveworkgroupmember
- **Description**  
Add or update a Workgroup Member. Specify Customer Record ID, Workgroup ID, and if Workgroup Manager.

# User Operations

## Delete a Batch of Users

- **URL**  
POST /api/V1/deleteuserbatch  
POST /api/V2/deleteuserbatch
- **Description**  
Delete a batch of Users. To get User record IDs, use "Get a User by login ID" or "Get a User by public ID."

## Delete a User by Record ID

- **URL**  
DELETE /api/V1/deleteuser/userrecordid/{userrecordid}  
DELETE /api/V2/deleteuser/userrecordid/{userrecordid}
- **Description**  
Delete a User by record ID. To get User record IDs, use "Get a User by login ID" or "Get a User by public ID."

## Get a List of Users

- **URL**  
GET /api/V1/getlistofusers
- **Description**  
Get a list of all system Users.

## Get a Batch of Users

- **URL**  
POST /api/V1/getuserbatch
- **Description**  
Get User information in a batch.

## Get a User by Login ID

- **URL**  
GET /api/V1/getuserbyloginid/loginid/{loginid}  
GET /api/V2/getuserbyloginid/  
GET /api/V3/getuserbyloginid/
- **Description**

Get detailed User information by login ID. Use to get User record IDs and account settings, for example.

### Get a User by Public ID

- **URL**  
GET /api/V1/getuserbypublicid/publicid/{publicid}  
GET /api/V2/getuserbypublicid/publicid/{publicid}
- **Description**  
Get detailed User information by public ID. Use to get User record IDs and account settings, for example.

### Get a User by Record ID

- **URL**  
GET /api/V1/getuserbyrecid/recid/{recid}
- **Description**  
Get detailed User information by record ID. Use to get user public IDs and account settings, for example.

### Create or Update a Batch of Users

- **URL**  
POST /api/V1/saveuserbatch  
POST /api/V2/saveuserbatch
- **Description**  
Create or update Users in a batch. To update, specify record ID. To create, leave record ID empty.

### Create or Update a User

- **URL**  
POST /api/V1/saveuser  
POST /api/V2/saveuser
- **Description**  
Create or update a User. The response is a collection because if you use a public ID, more than one User could be updated since public IDs may not be unique.

# Parameter Glossary

The following conventions are used in this glossary:

- **Internal ID:** The database ID for a specific object.
- **Name:** Typically, the display name for an object.
- **Public ID:** Typically, the ID shown to users, such as an incident ID or a user's full name.
- **Record ID:** The database ID for a specific record.
- **Search Item:** Saved searches, also known as a stored query.
- **Search Results:** A set of records returned by a Search.

The operation usage is noted for each parameter, unless a parameter is used across multiple operation types, such as *busobid*.

- ***.expires***

For Token operation responses, the date and time a token expires.

- ***.issued***

For Token operation responses, the date and time a token was issued.

- ***access\_token***

For Token operation responses, the issued access token.

- ***accountLocked***

For Security operations, a flag to indicate that a user account is locked.

- ***actions***

For Mobile Form operation responses, a list of action objects available for a Form.

- ***add***

For Security operation responses, a flag to indicate if "Add" is granted to a Security Group for a specific right.

- ***addUserToTeamRequests***

For Security operation requests, a list of objects for adding users to Teams.

- ***allfields***

For Business Object operation requests, a flag to include all related Business Object Fields.

- ***allow***

For Security operation responses, a flag to indicate if "Allow" is granted to a Security Group for a specific right.

- ***allowQuickSearch***

For Quick Search operation responses, a flag to indicate if Quick Search is allowed.

- ***allowSpecificSearch***

For Quick Search operation responses, a flag to indicate if Specific Search is allowed.

- ***allowValuesOnly***

For Search operation responses, a flag to indicate that only specified values can be used for a prompt (true) or that any value can be provided (false).

- ***altitude***

For Mobile Form operation responses, the altitude for a specific Business Object record (if location awareness is enabled).

- ***apiVersion***

For Service Information responses, the version of the Cherwell REST API you are using.

- ***applicationtype***

For Security operation requests, the type of application to get authentication settings for. Values are RichClient (CSM Desktop Client), BrowserClient, BrowserPortal (CSM Portal), and MobileClient (Cherwell Mobile).

- ***ascending***

For Quick Search operation responses, a flag to indicate results are sorted in ascending order.

- ***as:client\_id***

For Token operation responses, API Client ID used to request a token.

- ***association***

The internal ID for the Business Object association.

- ***attachBusObId***

For Business Object attachment requests, the internal ID for the type of Business Object to attach to another Business Object.

- ***attachBusObName***



For Business Object attachment requests, the display name for the type of Business Object you want to attach to another Business Object.

- ***attachBusObPublicId***

For Business Object attachment requests, the public ID for the type of Business Object you want to attach to another Business Object.

- ***attachBusObRecId***

For Business Object attachment requests, the internal ID for the Business Object to attach to another Business Object.

- ***attachedBusObId***

For Business Object attachment responses, the internal ID for the type of Business Object attached to another Business Object.

- ***attachedBusObRecId***

For Business Object attachment responses, the internal ID for the Business Object attached to another Business Object.

- ***attachmentFileId***

For Business Object attachment operations, the internal ID for a file attachment.

- ***attachmentFileName***

For Business Object attachment operations, the path and file name of an file attachment.

- ***attachmentFileType***

For Business Object attachment operations, the type of file attachment.

- ***attachmentId***

For Business Object attachment operations, the internal ID for a record that contains information about a file attachment.

- ***attachments***

For Business Object attachment operations, a list of objects related to Business Object attachments.

- ***attachmentTypes***

For Business Object attachment operations, the type of file attachment:

- 0 = Imported into the database.
- 1 = Linked to an external file.
- 2 = URL attachment.

- **attributes**

A list of attributes defined for specific Fields.

- **auth\_mode**

For Token operation requests, the authentication mode to use for requesting an access tokens. Options are Internal (for CSM authentication), Windows, LDAP, and SAML.

- **autoFill**

For field schema responses, a flag that indicates if Auto-Populate is enabled.

- **body**

For Business Object attachment upload operations, the body of the request is the byte array of the file part being uploaded.

- **browserClientCustomViewId**

For Security operation responses, the internal ID of the Browser Client View assigned to a Role.

- **businessObjectExcludeList**

For Security operation responses, the internal IDs for Business Objects that are excluded for a Role.

- **busobid/busObID**

The internal ID for a Business Object type, such as Incident or Task.

- **busobname/busObName**

The display name for a Business Object type. Examples include:

- Incident
- Task
- Service Cart

- **busobrecid**

The internal ID for a single Business Object record.

- **calculated**

For field schema responses, a flag that indicates if an Expression is used to calculate the field's value.

- **caption**

For field schema responses, the title for a column on the Business Object Grid.

- **category**

For field schema responses, a string that indicates the folder used to organize the field.

- ***categoryDescription - getsecuritygroupcategories***

Not used by the REST API.

- ***categoryid***

For Security Group operations, the internal ID for the Security Group category.

- ***categoryname***

For Security Group operations, the display name for the Security Group category.

- ***changedLimits***

For Quick Search operation responses, the limits based on changes made in a specific time frame.

- ***changedOption***

For Search Item operations, an indicator that a change limit can be used. Use the Option Key to determine if you can change this option.

Option Key:

- 0 = None (Not selected and cannot select.)
- 1 = Use (Selected and cannot clear.)
- 2 = Display (Not selected and can select.)
- 3 = UseAndDisplay (Selected and can clear.)

- ***childFolders***

For Search Item operation responses, a list of child folders that contain Stored Queries.

- ***childItems***

For Search Item operation responses, a list of Stored Queries that are children of a scope.

- ***client\_id***

For Token operation requests, the API client ID for the client making the token request.

- ***client\_secret***

For Token operation requests, the API client secret for the native client making the token request. This is only required for native clients.

- ***columns***

For Quick Search operation responses, a list of properties for each field on the Business Object Grid. Columns are returned when the schema flag is set to true.

- ***comment***  
A text string.
- ***constraintXml***  
For Search Results operation responses, an XML version of properties for a constraint used for a Prompt.
- ***created***  
For Business Object attachment operations, the date and time an attachment was added to a Business Object.
- ***createDateTime***  
For Security operation responses, the date and time a user account was created.
- ***csmCulture***  
For Service Information operations, the default culture for CSM.
- ***csmVersion***  
For Service Information responses, the CSM version.
- ***culture***  
For Security operation responses, the language code for the culture assigned to a Role.
- ***cultureCode***  
For Core operations, the internal code for a culture.
- ***currencyCulture***  
For field schema responses, the Currency Culture set for Number fields.
- ***currencySymbol***  
For field schema responses, the currency symbol for the Currency Culture set for Number fields.
- ***customGridDefId***  
For Search operations, the internal ID for the Grid used to override Field list settings.
- ***customGridId***  
For Business Object operations, the internal ID for the Grid used to override Field list settings.
- ***customSeparator***

For Business Object attachment requests, a custom separator for exports to a text file.

- ***dateTimeFormatting***

For ExportSearchResults only, the format for Date/Time fields. For example: mm/dd/yyyy or dd/mm/yyyy.

- ***decimalDigits***

For field schema responses, the number of digits for a Number field that appear in the field value after the decimal point.

- ***default***

For Search Results operation responses, the default value set for a Prompt.

- ***defaultSortOrderAscending***

For Search operation responses, a flag to indicate the default sort order. True is ascending; false is descending.

- ***defaultToQuickSearch***

For Quick Search operation responses, a flag to indicate that if both Quick Search and Specific Search are allowed, Quick Search is the default (true).

- ***delete***

For Security operation responses, a flag to indicate if "Delete" is granted to a Security Group for a specific right.

- ***deleteRequests***

A list of deleteRequest objects that will be processed.

- ***departmentMemberEdit***

For Security operation responses, a flag to indicate if a department member can edit Business Object records or fields in those records.

- ***departmentMemberView***

For Security operation responses, a flag to indicate if a department member can view Business Object records or fields in those records.

- ***description***

The description of an object.

- ***details***

For field schema responses, a string of attributes set for a field.

- ***dirty***

A flag to update a Field value if true. If false, the value is not updated.

- ***displaytext/displayName***

The display text for the item.

For example, use the attachment name, which is the display text for an attachment record.

- ***docRepositoryItemID***

For Quick Search operation responses, the internal ID for a Document Repository.

- ***edit***

For Security operation responses, a flag to indicate if Edit rights are granted.

- ***emailAlias***

For Security operation requests, the e-mail alias assigned to a Team.

- ***enabled***

For field schema responses, an expression that is being evaluated.

- ***error***

An error message. (Deprecated in V2 operations.)

- ***errorCode***

A text string used to indicate a type of error message.

- ***errorMessage***

An error message.

- ***expires\_in***

For Token operation responses, the number of seconds before a token expires.

- ***exportFormat***

For Search Operation requests, the format of exported searches:

- 0 = CSV
- 1 = Excel
- 2 = Tab

- 3 = Word
- 4 = Custom Separator
- 5 = Simple JSON

- ***exportTitle***

Not applicable to the REST API.

- ***fieldDefinitions***

For field schema responses, a list of properties for each field.

- ***fieldId***

The internal ID for a Field.

- ***fieldName***

The display name of a field.

- ***fieldNames***

For Business Object template requests, a comma-delimited list of Field names. By default, all Fields are requested.

- ***fieldPermissions***

For Security operation responses, a list of rights objects for a Security Group.

- ***fields***

In most cases, set a Field parameters, such as *fieldId*, *name*, and *value*.

For search operations, a list of Fields to return.

- ***fieldsList***

A specific list of Fields to request.

- ***fieldValidationErrors***

A list of objects related to field validation.

- ***filename***

For Business Object attachment requests, the name of a file attachment being uploaded.

- ***filters***

For Search operations, a set of filters to apply to searches. Each filter includes fieldId, operator, and value.

- ***firstReclIdField***

For Business Object schema operations, the first Record ID field specified for a Business Object.

- ***folder/folderId***

The name or internal ID of an item's folder.

- ***foredit***

For Form operations, a flag to get an editable version of a Form.

- ***formId***

The internal ID for a Form.

- ***fullFieldID***

A string containing the internal Business Object ID and internal Field ID.

- ***galleryImage***

For Mobile Form operations, the image shown on the form.

- ***grant\_type***

For Token operation requests, the type of token being requested: password or refresh token.

- ***gridDefinitions***

The internal IDs for grids associated with a Business Object.

- ***gridId***

The internal ID for the custom Grid that contains the Field list.

- ***group***

For Business Object operation responses, a flag to indicate that the Business Object is a group leader.

- ***groupid***

For Security operations, the internal ID for a Security Group.

- ***groupname***

For Security operations, the display name for a Security group.



- **groups**

For Quick Search operation responses, a set of objects and a results list for a simple text search.

- **groupSummaries**

For Business Object operation responses, a list of objects for members of a Business Object group.

- **hasAnyOptions**

For Quick Search operation responses, true if search configuration option is set to Display (2) or UseAndDisplay (3).

Option Key:

- 0 = None (Not selected and cannot select.)
- 1 = Use (Selected and cannot clear.)
- 2 = Display (Not selected and can select.)
- 3 = UseAndDisplay (Selected and can clear.)

- **hasDate**

For field schema responses, indicates if a Date/Time field displays dates.

- **hasDefaultSortField**

For field schema responses, a flag to indicate if a field has a default sort order. Use with *defaultSortOrder*.

- **hasError**

A flag to indicate an error needs to be reviewed.

- **hasPrompts**

For Search Result operation responses, a flag that indicates that a Stored Query has prompts.

- **hasTime**

For field schema responses, indicates if a Date/Time field displays times.

- **height**

For Get Gallery Image operation requests, the height of an icon.

- **historyNotes**

For Queue operation requests, notes to be added to the history log for the item.

- **historyReclId**

For Queue operation responses, the record ID of the Business Object with the history.

- ***historyText***

For Queue operation responses, the text of the history record for the item.

- ***historyTypeid***

For Queue operation responses, the ID of the Business Object type that holds the history.

- ***html***

For Business Object operation responses, the placeholder for the HTML field value. It is only populated on getting field values for rich text fields.

- ***id***

For Mobile Form operations, the internal ID for actions available for the Form.

- ***image***

For Security operation requests, the image assigned to a Team.

- ***includeAll***

For Business Object operation requests, a flag to include all Fields with Business Object templates.

- ***includeAllFields***

For Search operation requests, a flag to include all Fields in ad-hoc search results.

- ***IncludeAvailableInSpecific***

For Quick Search operation responses, flag to indicate whether available Quick Search items are included.

- ***includelinks***

A flag to request hyperlinks in results. Default is false.

- ***IncludeQuickSearchInSpecific***

For Quick Search operation responses, a flag to include all items in Quick Search in Specific searches.

- ***includerelationships***

For Business Object operation requests, a flag to include schemas for related Business Objects. Default is false.

- ***includeRequired***

For Business Object operation requests, a flag to request all required Fields with Business Object templates.

- ***includeSchema***

For Search operations, a flag to include the table schema of the saved search. If false, results contain the fieldid and Field value without Field information. Default is false.

- ***internalLoginAllowed***

For Security operations, a flag to indicate if internal authentication is configured for a specified client.

- ***isBinary***

For field schema responses, a flag to indicate that field contains Binary data, such as an image.

- ***isBusObTarget***

For Quick Search operation responses, a flag to indicate that the Quick Search is based on a Business Object (true) or a Document Repository (false).

- ***isCurrency***

For field schema responses, a flag to indicate that a Number field stores currency values.

- ***isDateRange***

For Search Results operation responses, a flag to indicate that a prompt includes a date range.

- ***isDateTime***

For field schema responses, a flag to indicate a Date/Time field.

- ***isFilterAllowed***

For field schema responses, a flag to indicate that filtering is allowed.

- ***isFullTextSearchable***

For field schema responses, a flag to indicate if full-text searching is enabled for a field.

- ***isLogical***

For field schema responses, a flag to indicate a Logical field.

- ***isNumber***

For field schema responses, a flag to indicate a Number field.

- ***isShortDate***

For field schema responses, a flag to indicate that a Date/Time field shows dates only.

- ***isYesNoRight***

For Security operation responses, a flag to indicate that a Security Group right is either allowed or not allowed.

- ***isVisible***

For field schema responses, a flag to indicate if a field is visible.

- ***key***

For Quick Search operation responses, the name of a field used for sorting.

- ***label***

For Mobile Form operations, the display name for Fields.

- ***lastPasswordResetDate***

For Security operation responses, the date a user's password was last reset.

- ***lastPasswordResetTime***

For Security operation responses, the time a user's password was last reset.

- ***latitude***

For Mobile Form operation responses, the latitude for a specific Business Object record (if location awareness is enabled).

- ***ldapLoginAllowed***

For Security operations, a flag to indicate if LDAP authentication is configured for a specified client.

- ***links***

For requests, a flag to request hyperlinks in results. Default is false.

For responses, a list of links, including name and URL.

- ***ldapRequired***

For Security operation requests, a flag to indicate whether Active Directory group membership is required.

- ***listDisplayOption***

For Search Results operation responses, the list display option for a Prompt:

- 0 = Auto

- 1 = Simple text box
- 2 = List of values in a combo
- 3 = List of values in a grid
- 4 = List of values in a simple list
- ***listReturnFieldId***

For Search Results operation responses, the internal ID for the field returned when a Prompt list is a Grid.
- ***localizedScopeName***

For Search Item operation responses, a translated scope name based on a user's assigned culture.
- ***locationInformation***

For Mobile Form operation responses, a list of location objects for a specific Business Object record.
- ***loginId***

A user's login ID.
- ***lookup***

For Business Object operations, a flag indicating a Business Object type of Lookup.
- ***longitude***

For Mobile Form operation responses, the longitude for a specific Business Object record (if location awareness is enabled).
- ***major***

For Business Object operations, a flag indicating a Business Object type of Major.
- ***managerOfOwnerEdit***

For Security operation responses, a flag to indicate if a manager of an owner can edit Business Object records or fields in those records.
- ***managerOfOwnerView***

For Security operation responses, a flag to indicate if a manager of an owner can view Business Object records or fields in those records.
- ***maximumSize***

For field schema responses for text fields, the maximum number of characters allowed.
- ***mobileClientCustomViewId***

Not applicable to the REST API.

- ***multiline***

For Mobile Forms, a flag to indicate if a field can span multiple lines.

For Prompts, a flag to indicate if a text prompt can contain multiple lines.

- ***name***

The display name for an object.

- ***nextPasswordResetDate***

For Security operations, the date on which users are required to change their passwords.

- ***nonFinalStateOption***

For Quick Search operation configuration responses, a flag to indicate if closed records are excluded. True excludes closed records. Use the Option Key to determine if you can change this option.

Option Key:

- 0 = None (Not selected and cannot select.)
- 1 = Use (Selected and cannot clear.)
- 2 = Display (Not selected and can select.)
- 3 = UseAndDisplay (Selected and can clear.)

- ***nonScopeOwnerAdd***

For Security operation responses, a flag to indicate that users are granted the "Add" right for items outside of their ownership scope.

- ***nonScopeOwnerDelete***

For Security operation responses, a flag to indicate that users are granted the "Delete" right for items outside of their ownership scope.

- ***nonScopeOwnerEdit***

For Security operation responses, a flag to indicate that users are granted the "Edit" right for items outside of their ownership scope.

- ***nonScopeOwnerView***

For Security operation responses, a flag to indicate that users are granted the "View" right for items outside of their ownership scope.

- ***offset***

For Business Object attachment upload operation requests, the starting index of the file part being uploaded. For the first part, the offset should be zero.

- ***operator***

A string used to evaluate filters.

- ***owner***

The internal ID for the Business Object association.

- ***ownerEdit***

For Security operation responses, a flag to indicate if an owner can edit Business Object records or fields in those records.

- ***ownerView***

For Security operation responses, a flag to indicate if an owner can view Business Object records or fields in those records.

- ***pageNumber***

The page number of the result set to return.

- ***pageSize***

The number of rows to return per page.

- ***parentbusobid***

The internal ID for the parent Business Object type.

- ***parentBusObPublicId***

The public ID for the parent Business Object.

- ***parentbusobrecid***

The record ID for the parent Business Object.

- ***parentFolderId***

For Search Item operation responses, the name or internal ID of a folder that contains Stored Queries.

- ***password***

The password assigned to a user account.

- ***passwordNeverExpires***

For Security operations, a flag to indicate whether a user's password expires or not.

- ***primaryBusObId***

For Security operation responses, the internal ID of the primary Business Object assigned to a Role.

- ***promptId***

For Search operations, the internal ID or name for a prompt.

- ***promptName***

For Search operations, the display name of a prompt.

- ***prompts***

For Search Results operation responses, a list of objects associated with Prompts.

- ***promptType***

For Search Results operation responses, the type of Prompt:

- 1 = Text
- 2 = Number
- 3 = Date/Time
- 4 = Logical
- 5 = Date
- 6 = Time

- ***promptValue***

For Search operation responses, the default value for a Prompt.

- ***promptValues***

For Search operation requests, a Prompt ID and value. This can be a list of multiple Prompts needed for each request.

For Search operation responses, a list of values to choose from. Values are not processed at runtime, however.

- ***publicid/publicId***

For Business Object operations, the public ID used to identify Business Object records.

For Security operations, the public ID used to identify users.

- ***queueStandInKey***

For Queue operations, the internal key for a queue derived from its scope, scope owner, and folder.



- ***quickSearchId***  
For Quick Search operation responses, the internal ID of a Quick Search configuration.
- ***quickSearchItems***  
For Quick Search operation responses, a set of object for each Quick Search configuration.
- ***quickSearchWatermark***  
For Quick Search operation responses, the text provided for the Quick Search control.
- ***readOnly***  
For field schema responses, a flag to indicate if the field is read only (true) or editable (false).
- ***readRequests***  
A list of readRequest objects used to return a batch of items, such as Business Objects or users.
- ***recIdFields***  
The record ID Fields specified for a Business Object.
- ***refresh\_token***  
For Token operations, The refresh token used to grant another access token.
- ***relatedBusinessObjects***  
For Business Object operations, a list of objects for related Business Objects.
- ***relationshipid***  
The internal Relationship ID for the related Business Object.
- ***required***  
A flag to indicate an item is required (true) or optional(false).
- ***resolvedQuickSearchWatermark***  
For Quick Search operation responses, the text provided for the Quick Search control. Custom text is shown if it was provided.
- ***rightId***  
For Security operation responses, the internal ID for a Security Group right.
- ***rightName***

For Security operation responses, the internal name for a right.

- ***roleId***

For Security operation responses, the internal ID for a Role.

- ***roleName***

For Security operation responses, the display name for a Role.

- ***roles***

For Security operation responses, a list of objects related to Roles.

- ***rowColor***

For Quick Search operation responses, the row color assigned to a record returned by the search.

- ***rows***

For Quick Search operation responses, a list or records returned based on search criteria.

- ***samlLoginAllowed***

For Security operations, a flag to indicate if SAML authentication is configured for a specified client.

- ***saveRequests***

A list of saveRequest objects that will be processed.

- ***scanCode***

The scan code for a specific Business Object record.

- ***scope***

The name or internal ID of an item's Scope. For example, *Global* and *Team* are Scopes.

- ***scopeowner***

The internal ID or name of the Scope Owner. For example, if *Team* is the Scope, *1st Level Support* might be the Scope Owner.

- ***searchAnyWordsOption***

For Quick Search operation configuration responses, a flag to indicate whether any or all words will be used in a query. Use the Option Key to determine if you can change this option.

Option Key:

- 0 = None (Not selected and cannot select.)

- 1 = Use (Selected and cannot clear.)
- 2 = Display (Not selected and can select.)
- 3 = UseAndDisplay (Selected and can clear.)

- ***searchAttachmentsOption***

For Quick Search operation configuration responses, a flag to indicate whether attachments will be queried. Use the Option Key to determine if you can change this option.

Option Key:

- 0 = None (Not selected and cannot select.)
- 1 = Use (Selected and cannot clear.)
- 2 = Display (Not selected and can select.)
- 3 = UseAndDisplay (Selected and can clear.)

- ***searchID***

For Search operations, the internal ID of a Stored Query.

- ***searchName***

For Search operations, the display name of a Stored Query.

- ***searchRelatedOptions***

For Quick Search operation configuration responses, a flag to indicate whether related Business Objects will be queried. Use the Option Key to determine if you can change this option.

- ***searchResultsFields***

For Search Result operation responses, a list of field schema objects.

- ***searchResultsFieldValues***

For Quick Search operation responses, a list of field value objects returned for a specific search. Objects are fieldId, name, value, and the dirty flag.

- ***searchTargetId***

For Quick Search operation responses, the Business Object ID or Document Repository to query.

- ***searchTargetType***

For Quick Search operation responses, either Business Object or Document Repository.

- ***searchTerm/searchText***

For Search operations, a text string used to filter search results.

- **SecurityGroupId**

For Security operations, the internal ID for a Security Group.

- **securityGroups**

For Security operation responses, a list of objects related to Security Groups.

- **sectionFields**

For Mobile Form operations, a list of Fields and Attributes for each section on a form.

- **sections**

For Mobile Form operations, a list of sections available on a Form.

- **selectedChangeLimit**

For Quick Search operation requests, the change time frame for the query. For example:

```
{
    "displayName": "Yesterday",
    "units": "Days",
    "value": -1
},
```

Possible values are based on the unit. In the example above, -1 for the Days unit equals today minus 1 day.

- **selectedSortByFieldId**

For Quick Search operation requests, the fieldID of a field used to sort results.

- **shortDisplayName**

For Security operation responses, the display name for a user.

- **simpleResultsListItems**

For Quick Search operation responses, a list of records returned for a query.

- **smartClientCustomViewId**

For Security operation responses, the internal ID of the Desktop Client View assigned to a Role.

- **sortable**

For field schema responses, a flag to indicate that a field is sortable (true) or not sortable (false).

- ***sortByFields***

For Quick Search operation configuration responses, a list of pre-defined options available for the search.

- ***sortByOption***

For Quick Search operation configuration responses, an option to indicate if sorting is allowed. Use the Option Key to determine if you can change this option.

Option Key:

- 0 = None (Not selected and cannot select.)
- 1 = Use (Selected and cannot clear.)
- 2 = Display (Not selected and can select.)
- 3 = UseAndDisplay (Selected and can clear.)

- ***sortDirection***

For Search operations, the possible values are:

- 0 = No sorting
- 1 = Ascending
- 2 = Descending

- ***sorting***

For Search operations, a set of objects used to sort search results.

- ***sortOrder***

For field schema responses, indicates sort order (ascending or descending).

- ***specificSearchItems***

For Quick Search operation configuration responses, a list of options and settings for specific Business Object Quick Searches.

- ***standardRightName***

For Security operation responses, the internal name for a right, prepended with the right's Category.

- ***standInKey***

For Core operations, the internal key for an item derived from its scope, scope owner, and folder.

- ***stateFieldId***

For Business Object operation responses, the internal ID for the State field defined for a Business Object life cycle.

- ***states***

For Business Object operation responses, the display names for State values defined for a Business Object life cycle.

- ***stopOnError***

A flag to stop or continue on error.

- ***storageName***

For field schema responses, the database name used for a Field.

- ***subtitle***

For Quick Search operation responses, the Last Modified Subtitle for a search result record.

- ***supportedAssociations***

For Search Item operation responses, a list of Business Objects associated with the Business Object you are querying.

- ***supporting***

For Business Object operations, a flag indicating a Business Object type of Supporting.

- ***systemDateTime***

For Service Information responses, the current date and time for the CSM server in UTC format.

- ***systemUtcOffset***

For Service Information responses, the UTC offset so the system can transform a date/time.

- ***targetId***

For Quick Search operation responses, the Business Object ID to query.

- ***targetBusObId***

For Mobile Form operations, the Business Object ID for the Business Object type.

- ***targetBusObRecId***

For Mobile Form operations, the Business Object record ID.

- ***teamEdit***

For Security operation responses, a flag to indicate if a Team member can edit Business Object records or fields in those records.

- ***teamId***

For Security operations, the internal ID for a Team.

- ***teamManagerOfOwnerEdit***

For Security operation responses, a flag to indicate if a Team manager can edit field in a Business Object record.

- ***teamManagerOfOwnerView***

For Security operation responses, a flag to indicate if a Team manager can view field in a Business Object record.

- ***teamName***

For Security operation responses, the display name for a Team.

- ***teams***

For Security operation responses, a list of objects related to Teams.

- ***teamType***

For Security operations, the type of Team:

- User = User Teams
- CustomerWorkgroup = Customer Workgroups

- ***teamView***

For Security operation responses, a flag to indicate if a Team member can view Business Object records or fields in those records.

- ***text***

For Search Results operation responses, the text shown in the Prompt window.

- ***timeZone***

For Service Information responses, the time zone of the CSM Application Server. Data returned is a serialization of the .NET TimeZoneInfo class.

- ***title***

For Mobile Form operations, the display name of a Business Object.

For Quick Search operation responses, the search criteria for a simple text search.

- ***token\_type***

For Token operations, the type of token. Currently, the type is always "bearer."

- ***totalRecords***

For batch delete operations, the total number of records deleted.

- ***totalRows***

For Search Results operation responses, the number of records returned by a search.

- ***totalsize***

For Business Object attachment operations, the size of a file, in bytes.

- ***type (Attachments)***

The type of record attachment:

- 0 = None: Not applicable to the REST API.
- 1 = File: Linked files.
- 2 = FileManagerFile: Imported files.
- 3 = BusOb: Attached Business Objects.
- 4 = History: Information about the attachment, if available.
- 5 = Other: Not applicable to the REST API.

- ***type (Business Objects)***

The type of Business Object:

- All
- Major
- Supporting
- Lookup
- Groups

- ***type (Fields)/fieldType***

The type of field (Text, Date/Time, etc.).

- ***uncFilePath***

For Business Object attachment operations, a file path to a Business Object linked attachment.

- ***url***

A URL string.

- ***usedefaultgrid***

A flag to trigger the use of the related Business Objects default grid for the list of Fields to return.

- ***userCannotChangePassword***

For Security operations, a flag to prevent users from changing their passwords.



- ***userInfoFields***

For Security operations, a list of objects related to user information Fields.

- ***userIsTeamManager***

For Security operations, a flag to indicate that a user is a manager of a Team.

- ***userMustChangePasswordAtNextLogin***

For Security operations, a flag to require users to change their password when they log in next.

- ***username***

A CSM user's Login ID.

- ***userrecordid/userRecID***

The internal ID for a CSM user.

- ***useSortBy***

For Quick Search operation responses, a flag to indicate if sorting should be used (true) or not (false).

- ***validated***

For field schema responses, a flag to indicate whether validation is defined for the field. See *details* for validation information.

- ***value***

A string used to identify a value or to filter results.

- ***view***

For Security operation responses, a flag to indicate if View rights are granted.

- ***viewRunOpen***

For Security operation responses, a flag to indicate if "Run," "View," or "Open" are granted to a Security Group for a specific right.

- ***watermarkText***

For Quick Search operation responses, the text provided for the Quick Search control for specific searches.

- ***wholeDigits***

For field schema responses, indicates the number of whole digits specified for number fields.

- ***width***

For Get Gallery Image operation requests, the width of an icon.

- ***windowsLoginAllowed***

For Security operations, a flag to indicate if Windows authentication is configured for a specified client.

- ***windowsUserId***

For Security operations, the Windows login ID for a CSM user.