

User create or update API

Last updated Wednesday, 21 October 2015





Contents

Overview	3
Authenticating the API request.....	4
Configuring API keys	4
Building Authentication Request	4
User Details	5
Basic User Details	5
User Custom Attributes.....	6
User Group Memberships.....	6
User Create or Update Options.....	7
Assumptions	8

Overview

Janison CLS has an API endpoint for creating and updating CLS users. This document will explain the API spec so it can be used by CLS clients. This document should be distributed with a C# sample application which calls this API. Please ensure you have the "Janison User API Sample.cs" file for reference.

The API request is made up of three parts:

- Authentication details
- User Details for the creation/update
 - Basic user details such as username, first name, last name, email address, etc
 - Custom attributes values associated with the user
 - Groups the user should have memberships in
- Options
 - Various create and update options

The above points are detailed in this document.

Authenticating the API request

Configuring API keys

1. Navigate in the CLS to: **Settings -> Authentication.**
2. Under **Janison APIs**, select **Generate new key** for the **Primary** and **Secondary** keys.

Authentication

Authentication Model

Janison APIs

Each Janison API client connecting to this tenant should be configured to use the primary key or the secondary key if the primary key fails validation. This is to allow keys to be reset independently without affecting client connectivity.

Copy the key below and paste into the client's configuration.

Primary Key:

Generate new key

Secondary Key:

Generate new key

Access

Save settings

 or

Cancel

Contextual Help

Building Authentication Request

1. To call the API, a correctly formatted http request must be sent to the API endpoint Url with a valid HMAC.
2. The following header fields are required: Authorization, Date and Method.
 - a. The 'Authorization' header field = "JanisonAPI <userid>:<GeneratedHash>"
 - i. <userid> = CLS Username of the calling user
 - ii. <GeneratedHash> = HMAC (SHA256) of <hashData> using either Primary or Secondary API keys
 1. To calculate the HMAC:
 - a. Convert the Base-64 encoded Key to a string
 - b. Encode the <hashData> as a UTF8 byte array
 - c. Compute the Hash of the data using the key
 - d. Convert the result to Base-64
 2. <hashData> = "GET\n<url>\n<userId>\n<timestamp>"
 - a. <url> = Report API endpoint url.
 - b. <userId> = CLS Username of the calling user
 - c. <timestamp> = Current timestamp in RFC1123 format.
 - b. The 'Date' header field = <timestamp>
 - i. <timestamp> = Current timestamp (must match timestamp above)
 - c. The 'Method' header field = "GET"
3. The API returns XML by default but by specifying a JSON content-type in the HTTP request it will return JSON.

User Details

Basic User Details

User details need to be posted in the following format:

postData += "&ImportData.UserDetails.<AttributeName>=<AttributeValue>" e.g.

postData += "&ImportData.UserDetails.Surname=Smith";

The list of supported attributes can be found below.

Attribute Name	Mandatory	Data Type	Description
Username	True	String	User's unique identifier. This will be used for logging into the system.
Givenname	True	String	
Surname	False	String	
MiddleName	False	String	
PreferredName	False	String	Preferred Name will be used throughout the system if specified (Except for official documents like certificates).
Title	False	String	
DateOfBirth	False	Date	Format: dd/mm/yyyy
Gender	False	String	Either Female, Male, or blank.
EmailAddress	False	String	
MainRoleId	False	String	A non-system (i.e. custom) role that the user holds as their main role
MainRoleOrgEntity	False	String	The scope of the main role assignment. Possible values: Tenant, Org, or OrgUnit
Password	False	String	
OrgId	False	String	Organisation Id and Organisation Unit Id are mandatory if no Default Organisation Unit has been set in Settings > User Settings > Default User Settings
OrgUnitId	False	String	Organisation Id and Organisation Unit Id are mandatory if no Default Organisation Unit has been set in Settings > User Settings > Default User Settings
OrgUnitName	False	String	Only used if OrgUnit does not exist and AllowOrgUnitCreation is enabled
IsActive	False	Boolean	For specifying if the user is active. Possible values: True, False, Yes, No, Y, N, Blank. A blank value will be interpreted as active.
HomePhone	False	String	
WorkPhone	False	String	
MobilePhone	False	String	
Address	False	String	
Address2	False	String	
Suburb	False	String	
State	False	String	
Postcode	False	String	
Country	False	String	
StudentNo	False	String	This field can be used as a secondary unique identifier for users (in addition to logon username). The API will match users on this field if a match isn't found using logon username.

User Custom Attributes

User custom attributes need to be posted in the following format:

```
postData += "&ImportData.UserDetails.CustomAttributeData[0].CustomAttribute=<custom attribute id>";  
postData += "&ImportData.UserDetails.CustomAttributeData[0].Value=<custom attribute value>"; E.g.
```

```
postData += "&ImportData.UserDetails.CustomAttributeData[0].CustomAttribute=user-custom-attrib";  
postData += "&ImportData.UserDetails.CustomAttributeData[0].Value=test";
```

If you need to post multiple custom attributes for a user increment the "CustomAttributeData[]" index value.

User Group Memberships

User group memberships need to be posted in the following format:

```
postData += "&ImportData.UserDetails.GroupData[0].GroupType=<Group Type Id>";  
postData += "&ImportData.UserDetails.GroupData[0].GroupId=<Group Id>"; E.g.
```

```
postData += "&ImportData.UserDetails.GroupData[0].GroupType=Sport";  
postData += "&ImportData.UserDetails.GroupData[0].GroupId=Soccer";
```

If you need to post multiple group memberships for a user increment the "GroupData[]" index value.

User Create or Update Options

There are a number of options available when calling this API to alter the behaviour of the API.

Options need to be posted in the following format:

```
postData += "&ImportData.Options.<OptionType>=<OptionValue>" e.g.  
postData += "&ImportData.Options.AllowGroupCreation=true";
```

Below are the list of available options:

Option Name	Data Type	Option Description
AllowBypassOfPasswordStrengthTest	Boolean	If enabled, the password strength test will not be run
InactivateOldGroupMemberships	Boolean	If enabled, any group memberships the user has which are not specified in the API request will be inactivated. This only applies to group types specified in the API request
EmailNewUsers	Boolean	If enabled, new users will receive a welcome email with customisable message
AllowGroupCreation	Boolean	If enabled, any groups specified in the API request but not found in the system will be created
AllowOrgUnitCreation	Boolean	If enabled, any org unit specified in the API request but not found in the system will be created
AllowUpdateUsers	Boolean	If enabled, existing user records are allowed to be updated
AllowCustomAttributeListItemCreation	Boolean	If enabled, any existing custom attributes of type 'list' can have list items created if they exist in the API request but not in the system (up to a maximum of 20)

Assumptions

4. The API can only create or update a single user at a time. There is no support for bulk creating or updating users