

CLS Enrolment APIs

V1.3

January 2016



Contents

1. Introduction	3
2. Authenticating the API Request	4
Configuring Role	4
Creating the Janison API User	5
Configuring API keys.....	5
Building Authentication Request	6
3. Create Enrolment API	7
4. Update Enrolment API	9
5. Sync Enrolments API	12

1. Introduction

- 1.1. This document details the APIs related to enrolment management. Technical details are provided for connecting and consuming these APIs.

Please note: At time of writing (Document version v0.1-1.3) the APIs were not developed yet. The proposed API URLs and parameters are subject to minor change in the way the request may be formatted, however, the overall inputs and outputs will remain the same. Once the APIs are developed this documentation will be updated and this message removed.

2. Authenticating the API Request

Configuring Role

The Janison APIs can only be authenticated by a user who has a specific Janison API role. This role must be created from scratch and granted the rights appropriate for the specific usage of the API.

1. Navigate in the CLS to: Settings -> Roles
2. From the actions menu select the Add Role menu option
3. Enter the required information (Applicable documents section can be ignored):

New Role

Name*	<input type="text" value="Janison Api Role"/>
Identifier*	<input type="text" value="JanisonApiRole"/>
Description	<input type="text" value="This role be used by external systems connecting to the CLS"/>
	<input type="checkbox"/> Main Role
	<input type="checkbox"/> Not In Use

New Role

Name*	<input type="text" value="Janison Api Role"/>
Identifier*	<input type="text" value="JanisonApiRole"/>
Description	<input type="text" value="This role be used by external systems connecting to the CLS"/>
	<input checked="" type="checkbox"/> Main Role
	<input type="checkbox"/> Not In Use


4. After saving the new role, the “Securable areas of this application” must be set according to what APIs need to be accesses. Janison Support or the project implementation team can assist with this step. The Enrolment APIs detailed in this document are enabled under ‘Web Services’ -> ‘Enrolments’.

Creating the Janison API User

The Janison APIs need to be authenticated as a user. This user must have their main role set to the Janison API role as detailed in the previous section.

New User

User Profile Picture



Select profile picture...

Title

First Name

Janison Api User

Middle Name

Last Name

Organisation

(default)

Org Unit

(default)

System

Role

Janison Api Role

Scope of this role*

All Organisations

Save User

and add another

or Cancel

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 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

API Role

Janison Api Role

x

▼

3. Please note that "API Role" attribute must also be selected at this point. This step ensures that only users with this role are allowed to authenticate with Janison API.

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> = 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 JSON.

3. Create Enrolment API

Description

This API allows creation of new enrolments. The CLS has a rule which states a user can only have a single *open* enrolment per catalogue item. If the API is asked to create an enrolment where the user has an open enrolment existing for the given catalogue item then the API will not create a duplicate open enrolment but instead return the GUID of the existing open enrolment (Response Status 200 OK).

URL

/api/enrolments/create/{version}

Parameters

- {version}: The API Version

API Versions

- 1: Initial and current version

HTTP Method

POST

JSON Request Parameters

createEnrolmentData

Attribute Name	Mandatory	Data Type	Description
username	True	String (255)	User's unique identifier. This will be used for logging into the system.
catalogueItemIdentifier	True	String (255)	Unique identifier of the catalogue item to enrol the user into.
dueDate	False	Date	The due date of the enrolment. Date format is dd/mm/yyyy.
comments	False	String (500)	Any comments to be noted against the enrolment. Not visible to learner. These comments will show as an audit message below the enrolment details admin view.

Example JSON request object

```
{  
  "username": <string>,  
  "catalogueItemIdentifier": <string:GUID>,  
  "dueDate": <string:Date>,  
  "comments": <string>  
}
```

Response Object

createEnrolmentResultData

Attribute Name	Mandatory	Data Type	Description
clsEnrolmentId	False	GUID	The GUID of the enrolment created in the Janison CLS. This will only be populated if the enrolment was successfully created or existing open enrolment found.
wasExisting	False	Boolean	This attribute will be only be present if the request is successful (200). Returns TRUE if the enrolment was retrieved. Returns FALSE if the enrolment was created.
message	True	String (500)	Description of the API call result

Return Status Codes

- 200: Success
 - createEnrolmentResultData.clsEnrolmentId will be populated with GUID of CLS Enrolment Id
 - createEnrolmentResultData.message will be set to "Success"
 - createEnrolmentResultData.wasExisting will be set to depending on the retrieval or creation of the enrolment
- 400: Bad Request – when passed data object is incorrectly formed
 - createEnrolmentResultData.clsEnrolmentId will be an empty guid
 - createEnrolmentResultData.message will describe the problem if that is possible.
- 403: Forbidden
 - Returns no response object
- 404: Username or Catalogue Item Id not found
 - createEnrolmentResultData.clsEnrolmentId will be an empty guid
 - createEnrolmentResultData.message will describe what entity could not be found
- 500: Unexpected error
 - createEnrolmentResultData.clsEnrolmentId will be an empty guid
 - createEnrolmentResultData.message will describe what error occurred

Create Behaviour

- This API will bypass any enrolment approval workflows set against the catalogue item
- This API will bypass any catalogue item prerequisite rules
- If an open enrolment already exists for the Username/Catalogue Item combination, then a Status Code 200 will be returned and the createEnrolmentResultData.clsEnrolmentId attribute will be populated with the existing enrolment id.
- All new enrolments will be assigned the 'Not Attempted' status

4. Update Enrolment API

Description

This API allows updating of existing enrolments. The API consumer must pass as parameter the Enrolment GUID for the CLS enrolment they wish to update.

URL

/api/enrolments/update/{version}

Parameters

- {version}: The API Version

API Versions

- 1: Initial and current version

HTTP Method

POST

JSON Request Parameters

updateEnrolmentData

Attribute Name	Mandatory	Data Type	Description
clsEnrolmentId	True	GUID	The CLS system identifier of the enrolment
dueDate	False	Date	The due date of the enrolment. Date format is dd/mm/yyyy.
score	False	Float	Score awarded for the enrolment
lessonStatus	False	String (9)	Supported values: {Completed, Cancelled}
completedDate	False	Date	
comments	False	String (500)	Any comments to be noted against the enrolment. Not visible to learner. These comments will show as an audit message below the enrolment details admin view.

Example JSON request object

```
{  
  "clsEnrolmentId": <string:GUID>,  
  "dueDate": <string:Date>,  
  "score": <string:Float>,  
  "lessonStatus": <string:LessonStatus>,  
  "completedDate": <string:Date>,  
  "comments": <string>  
}
```

Response Object

updateEnrolmentResultData

Attribute Name	Mandatory	Data Type	Description
clsEnrolmentId	True	GUID	The GUID of the CLS enrolment
message	True	String (500)	Description of the API call result

Return Status Codes

- 200: Success
 - createEnrolmentResultData.enrolmentId will be populated with GUID of CLS Enrolment Id
 - createEnrolmentResultData.message will be set to "Success"
- 400: Bad Request – when passed data object is incorrectly formed
 - createEnrolmentResultData.enrolmentId will be empty
 - createEnrolmentResultData.message will describe the problem if that is possible.
- 403: Forbidden
 - Returns no response object
- 404: ClsEnrolmentId not found
 - createEnrolmentResultData.enrolmentId will be populated with GUID of CLS Enrolment Id
 - createEnrolmentResultData.message will describe that the enrolment could not be found
- 500: Unexpected error
 - createEnrolmentResultData.enrolmentId will be populated with GUID of CLS Enrolment Id
 - createEnrolmentResultData.message will describe what error occurred

Update behaviour

- If an enrolment is already completed or in any other *finalised* state then no update will take place. Completed, cancelled, and failed are all finalised states. A 500 error status code will be returned in this case.
- If no Lesson Status is specified then no attempt will be made to update the lesson status of the enrolment.
- If a lesson status is specified as “Completed” and no “Completed Date” attribute is specified then a 500 error status code will be returned
- If a lesson status other than “Completed” or “Cancelled” is specified then a 500 error status code will be returned
- Completing or cancelling an enrolment will not cascade complete or cancel learning assets associated with the enrolment.

5. Sync Enrolments API

Description

This API allows synchronising enrolments between the CLS and consumer. The consumer passes an array of “current” enrolments and the API will then determine which enrolments need to be created, updated, or cancelled. This batch approach can only be executed for one user per API call i.e. batch synchronising a group of users within one API call is not supported.

URL

/api/enrolments/sync/{version}

Parameters

- {version}: The API Version

API Versions

- 1: Initial and current version

HTTP Method

POST

JSON Request Parameters

syncEnrolmentRequestData

Attribute Name	Mandatory	Data Type	Description
username	True	String (255)	User's unique identifier. Sync Enrolments can only be called for a single user at a time.
enrolmentData	True	Array of syncEnrolmentData	An array of “current” enrolments as syncEnrolmentData objects (0 to 50 objects).
options	True	syncEnrolmentOptions	See details of the syncEnrolmentOptions object

syncEnrolmentData object

Attribute Name	Mandatory	Data Type	Description
consumerEnrolmentId	True	String (255)	The unique enrolment id as stored in the consumer system. This will be used when building the response object to identify the process result for the enrolment ONLY. This Id will not be stored in the CLS database.
clsEnrolmentId	False	GUID	If populated then the instance is interpreted as an UPDATE command. If

			empty then interpreted as CREATE command.
catalogueItemIdentifier	False	String (255)	Unique identifier of the catalogue item to enrol the user into.
dueDate	False	Date	The due date of the enrolment. Date format is dd/mm/yyyy.
score	False	Float	Score awarded for the enrolment
lessonStatus	False	String (9)	Supported values: {Completed, Cancelled}
completedDate	False	Date	
comments	False	String (500)	Any comments to be noted against the enrolment. Not visible to learner. These comments will show as an audit message below the enrolment details admin view.

syncEnrolmentOptions object

Attribute Name	Mandatory	Data Type	Description
allowCreate	False	Boolean	Defaults to False
allowUpdate	False	Boolean	Defaults to False
deactivateEnrolmentRule	False	Integer	Defaults to 0 0: "None" : No enrolments will be deactivated 1: "CancelEnrolments" : Any open enrolment for the given user that is present in the CLS but not in the array of syncEnrolmentData objects for "SyncEnrolments" request will be cancelled (Lesson Status set to cancelled for enrolment entity only). 2: "DueDateSetToNow" : Any open enrolment for the given user that is present in the CLS but not in the array of syncEnrolmentData objects for "SyncEnrolments" request will have their due date set to NOW. Using this rule assumes functionality described in section 4 is in place.

Example JSON request object (syncEnrolmentsData)

```
{
  "username": <string>,
  "enrolmentData": [ { <syncEnrolmentData> } ],
  "options": { <syncEnrolmentOptions> }
}
```

Response Object

syncEnrolmentResponse

Attribute Name	Mandatory	Data Type	Description
creates	True	Array of syncEnrolmentResultData	This attribute will be only be present if the request is successful (200). An array of created enrolments as syncEnrolmentResultData objects.
updates	True	Array of syncEnrolmentResultData	This attribute will be only be present if the request is successful (200). An array of updated enrolments as syncEnrolmentResultData objects.
deactivations	True	Array of syncEnrolmentDeactivationData	This attribute will be only be present if the request is successful (200). An array of deactivated enrolments as syncEnrolmentDeactivationData objects.
errors	True	Array of syncEnrolmentResultData	An array of enrolments that errored as syncEnrolmentResultData objects.
message	True	String (500)	Description of the API call result

syncEnrolmentResultData object

Attribute Name	Mandatory	Data Type	Description
consumerEnrolmentId	False	String (255)	The unique enrolment id as stored in the consumer system. This will only be blank in the case on an enrolment deactivation error.
clsEnrolmentId	False	GUID	The GUID of the CLS enrolment. When this object represents an error, this may be empty.
message	True	String (500)	Description of the sync enrolment result

syncEnrolmentDeactivationData object

Attribute Name	Mandatory	Data Type	Description
catalogueItemIdentifier	True	String (255)	The unique identifier of the CLS catalogue item.
clsEnrolmentId	True	GUID	The GUID of the CLS enrolment

Example JSON response object (syncEnrolmentsResultData)

```
{
  "creates": [{ <syncEnrolmentResultData> }],
  "updates": [{ <syncEnrolmentResultData> }],
  "deactivations": [{ <syncEnrolmentDeactivationData> }],
  "errors": [{ <syncEnrolmentResultData> }],
}
```

```
"message": <string>
```

Return Status Codes

- 200: Success
 - When all Creates, Updates and Deactivations have succeeded.
 - syncEnrolmentsResultData.create will contain array of created enrolments.
 - syncEnrolmentsResultData.update will contain array of updated enrolments.
 - syncEnrolmentsResultData.deactivations will contain array of deactivated enrolments.
 - syncEnrolmentsResultData.errors will be empty.
- 400: Bad Request – when passed data object is incorrectly formed
 - syncEnrolmentsResultData.message will describe the problem if that is possible.
- 403: Forbidden
 - Returns no response object
- 500: Unexpected error
 - No enrolment data provided
 - One or more enrolments failed. No enrolment updates are committed.
 - syncEnrolmentsResultData.errors will contain errors (if possible).
 - syncEnrolmentsResultData.message will describe the problem if that is possible.

Sync Behaviour

- The Sync Enrolments API will not accept more than 50 enrolments in the array of syncEnrolmentData request object. This limit is in place to prevent performance issues.
- The Sync Enrolments API will not accept 50 enrolments in the array of syncEnrolmentData request object if the deactivateEnrolmentRule is set to 1 or 2.
- If deactivateEnrolmentRule is set, enrolments that are in a finalised state (Completed, Cancelled, Failed) will not be affected, only open enrolments can be affected but this rule.
- If an UPDATE command is required and the enrolment to update is not linked to the specified Username then a 500 error will be returned with description of the error.