# **Coursera for Business: SuccessFactors LMS Integration**

# Introduction

**Summary**

This document explains how integration between Coursera for Business (C4B) and SAP SuccessFactors Learning Management System (LMS) is done from the technical perspective. The goal of the integration is to let the distributed platforms exchange data to have a consistent state, specifically:

* Courses available on the organizations C4B program should also be listed in SuccessFactors LMS
* Users whitelisted/blocked to access the organizations Coursera’s courses in LMS should be invited/removed to/from corresponding learning program in Coursera.
* Learners progress (statistics) calculated from the organizations C4B platform should also appear in LMS.

Technical intro

### **Communication API**

Both Coursera and LMS APIs are OAuth protected resources exposed by the platforms to exchange data between each other. In Coursera Program API, Content API, Invitation API, Status completion API and similarly in LMS Post Courses, Status completion APIs are OAuth protected thus every request to process or Post data should be accompanied with OAuth token which is validated against their respective Authorization Servers.

**Schedule**

Integration tool will Fetch and Post data at an agreed-upon time between the LMS and Coursera. A schedule can be changed if there is a need (after a prior agreement between parties).

### **Admin App**

This provides a user interface to enter LMS details which will be used by this middleware integration client to Get/Post data to the end applications i.e. Coursera and LMS.

#### 

#### **Actions:**

1. Client organization enter client specific details to integration client using admin App
2. Admin app initiates to get Auth code authorization grant from Coursera Authorization server to get the Auth token to access OAuth secured Coursera Business APIs
3. Admin App request for Auth Token and refresh token using Auth code in above step using Coursera Auth Server token Url.
4. Admin App saves these tokens in database so that batch jobs can use it while accessing Coursera Business APIs
5. When the Auth token expires Batch job needs to get the new token regenerated using token URL with refresh token grant type
6. **If refresh\_token is not permanent, when Coursera responds with invalid refresh token, job need to follow from step#B**

### **Order of integration batch jobs**

The batch jobs processing is executed in the following order:

1. Courses sync job

2. Invitation sync job

3. Course completion status sync job

# **Batch Jobs specifications:**

## **1.** **Courses Sync Job**

### **Introduction**

This job fetches Content data from Coursera Content Business API and publishes this data to the database. As all the Coursera APIs/ resources are OAuth protected, Course sync job in Client application needs to obtain the Auth Token. Each data request call from Client to Coursera Content API should send the token to authorize the request against the Coursera Authorization server. This ensures only valid request to access Content API data is catered and response returned to Authorized Client’s Course sync job.

In the next step Batch Job will post the fetched content data to LMS Courses API. Since this API is also OAuth protected, Client needs to obtain LMS Auth token from the LMS Auth server before the course sync job is posted to the LMS successfully.

#### **Actions**

1. An event triggers the Course sync batch job of the client application at the agreed upon schedule time. Course sync job launches the job and obtain Auth token from the Coursera Authorization server.
2. Batch Job sends the request to Coursera Program business API with the Auth token. Coursera Business API gets the request validates by authorizing token against its proprietary Authorization server and on successful validation returns response with content data to the client job.
3. Batch job send request to obtain the LMS Auth token from LMS Auth server.
4. Batch Job then prepares the OCN Course contents JSON required at the LMS side (mentioned in OCN document). Job prepares this OCN Courses JSON with multiple content chunks (configurable size) and post this Course contents JSON data to LMS Courses APIs
5. On successful validation of Auth token, LMS process the courses posted by the client job and returns the success response to the client Job.

**Courses sync job input data:**

|  |  |  |
| --- | --- | --- |
| Coursera Content API JSON fields | Field type | Field information |
| contentId | String | unique Id of content |
| name | String | Name of the course |
| description | String | Detailed description of the offered course |
| contentUrl | String | Launch url to the offered course |
| photoUrl | String | Course display pic url which will be displayed in LMS on successful course posting. This helps users recognize the course or series of courses |
| programId | String | Program Id under which this course is categorized. e.g. Data science/ Data Engineer, machine Learning etc. |

**Courses sync job output data:**

|  |  |  |
| --- | --- | --- |
| LMS Post courses API json fields | Field Type | field information |
| content[ ].contentID | String | Part of the content array. The unique ID for the section of content |
| content[ ].contentTitle | String | Part of the content array. This is the title for the section of the content. For example, the title for the course can be Artificial Intelligence and Creativity, but the section titles can be Defining Creativity, Can you Create Creativity? and so on |
| content[ ].launchType | String | Part of the content array. This is one of the allowed launch types. Fixed to 2. |
| description | String | Descriptions for the course. Each member of the array is the description in a language. For example |
| content[ ].launchURL | String | Part of the content array. This is the URL that users would click to launch this part of the course |
| thumbnailURI | String | A URI to an image that is the icon for the course. Thumbnails help users recognize the course or series of courses |
| courseID | String | This is the OCN provider's unique ID for the course. It is required so that SAP SuccessFactors Learning can maintain a map between learning items and OCN courses. |
| providerID | String | This is the unique ID of the provider as set in SAP SuccessFactors Learning. It is required. This is fixed as “COURSERA” |
| status | String | Part of the schedule array. It is a Boolean that indicates availability. If true, then the course is currently available. This is fixed as “ACTIVE” |

## 

## **2.** **Invite Sync Job**

## **Introduction**

### 

The Invite sync job is consists of 2 steps:-

1. Program Sync step to get programs from C4B API and save it into the database.
2. Invite Sync step to send the invite to the users against above fetched programs.

### **A. Program Sync Step**

This job fetches Programs data from Coursera for Business Program Business API and publishes this data to the database. As all the Coursera APIs/ resources are OAuth protected, program sync job in Client application needs to obtain the Auth Token. Each data request call from Client to Coursera Program API should send the token to authorize request against the Coursera Authorization server. This ensures only valid request to access Program API data is catered and response returned to Authorized Client’s program sync job.

**Actions**

1. An event triggers the Program sync job at the agreed upon schedule time in the client application. Program sync job launch the job and obtain Auth token from the Coursera Authorization server.
2. Program Job send request to Coursera Program business API with the Auth token. Coursera Business API validates the request by authorizing token against its proprietary Authorization server and on successful validation returns response with Program data to the client job.
3. Program Job receives program data. Process it and publish the data to Database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Coursera Program API json fields | Field type | Program Sync Job field name | Database column name | Field Information |
| Id | String | courseID | id | This is the course Id/ program Id |
| name | String | title | title | This field carries the Program name |
| url | String | launchURL | launch\_url | This is the launch URL of the program |
|  |  | providerID | provider\_id | This is the provider ID of the programs which publish data at the LMS site this is fixed as “COURSERA” |
|  |  | status | status | This is the course status. Its value is “ACTIVE” |

### **B. Invite Sync Step**

This job takes user data from the SFTP server and invites users to the Coursera program or deletes pending program invitation based on the user status. As all the Coursera APIs/ resources are OAuth protected, Invite sync job in Client application needs to obtain the Auth Token. Each request call from Client to Invite API should send the token to authorize request against the Coursera Authorization server. This ensures only valid request to post user data and response returned to Authorized Client’s Invite sync job.

#### **Actions**

1. An event triggers the Invite sync job at the agreed upon schedule time in the client application. Invite job first, fetch the user data file from the SFTP server. Below are the specifications of user file.

**File format**

It’s a CSV file with a comma (,) as a delimiter. All text values must be quoted with a double quote (“).

**Feed Information**

SSF-LMS will generate a feed that includes users information to whom program invitation needed to be sent. For every valid row in the feed, Invite job will add the user to the program’s invitation list.

**Inbound File Format**

Creator: SSF-LMS

File-Location: /uploads/dev/Inbound/

Filename: inbound.CSV

**Fields Format**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field type** | **Field information** |
| external ID | String | LMS specific identifier |
| first name | String | First name of the user |
| last name | String | Last name of the user |
| email | String | Email Id of the user |
| status | String | Status of the user(active or inactive) |

(b) Job obtain Auth token from the Coursera Authorization server and then sends post request to Coursera Invite business API along with the obtained Auth token.

(c) Coursera Business API validates the request by authorizing token against its proprietary Authorization server and on successful validation returns response to the client job.

**Invite sync job input data:**

|  |  |  |
| --- | --- | --- |
| Coursera Invite API json fields | Field type | Field information |
| externalId | String | LMS specific identifier |
| fullName | String | Full name of the user |
| email | String | Email Id of the user |

Invite sync job output data:

|  |  |  |
| --- | --- | --- |
| Coursera Invite API response json fields | Field type | Field information |
| id | String | Combination of programId and externalId |
| programId | String | This is the id of the program for which user is invited. |
| externalId | String | LMS specific identifier |
| fullName | String | Full name of the invited user |
| email | String | Email of the invited user |

## 

## 

## 

## 

## **3.** **Course completion status sync job**

## 

### **Introduction**

This job fetches course completion status data from Coursera course completion status Business API and publishes these data to the database. As all the Coursera APIs/ resources are OAuth protected, Course completion status sync job in Client application needs to obtain the Auth Token. Each data request call from Client to Coursera course completion status API should send the token to authorize request against the Coursera Authorization server. This ensures only valid request to access Course completion status API data is catered and response returned to Authorized Client’s Course completion status sync job.

In the next step Batch Job will post the fetched course completion status data to LMS course completion status API. Since this API is also OAuth protected, Client needs to obtain LMS Auth token from the LMS Auth server before the course completion status sync job is posted to the LMS successfully.

### 

#### **Actions**

(a) An event triggers the Course completion status sync batch job of the client application at the agreed upon schedule time. Course completion status sync job launches the job and obtain Auth token from the Coursera Authorization server.

(b) Batch Job sends the request to Course completion status business API with the Auth token. Coursera Business API gets the request validates by authorizing token against its proprietary Authorization server and on successful validation returns response with course status data to the client job.

(c) Batch job send request to obtain the LMS Auth token from LMS Auth server.

(d) Batch Job then prepares the OCN Course completion status JSON required at the LMS side (mentioned in OCN document). Job prepares this OCN Courses completion status JSON with multiple course status chunks (configurable size) and post this Course completion status JSON data to LMS Courses APIs.

(e) On successful validation of Auth token, LMS process the courses completion status posted by the client job and returns the success response to the client Job.

**Course completion status sync job input data:**

|  |  |  |
| --- | --- | --- |
| Coursera Content API json fields | Field type | Field information |
| id | String | Unique identifier of the entry. It has three parts, separated by “~”:  externalId, contentType, and contentId |
| contentId | String | Unique identifier for a “Course”. |
| externalId | String | Unique external identifier for a user. |
| contentType | String | The contentType will always be “Course”. |
| programId | String | Program Id under which this course is categorized. e.g. Data science/ Data Engineer, machine Learning etc. |
| enrolledAt | String | Timestamp for user enrollment in the course. |
| isCompleted | Boolean | A course is marked as completed when the user has finished all graded assignments. |
| completedAt | Date | Timestamp for user enrollment in the course. |
| grade | String | A number between 0 and 1.0 that represents the user’s grade for  the course. This field is only present when isCompleted is true and contentType is “Course”. |

**Course completion status sync job output data:**

|  |  |  |
| --- | --- | --- |
| LMS Post courses API json fields | Field Type | field information |
| userID | String | This is the SAP SuccessFactors Learning user ID for the user who completed the course. |
| courseID | String | This is the OCN provider's unique ID for the course. It is required so that SAP SuccessFactors Learning can maintain a map between learning items and OCN courses. |
| providerID | String | This is the unique ID of the provider as set in SAP SuccessFactors Learning. It is required. This is fixed as “COURSERA” |
| courseCompleted | Boolean | This is a Boolean. It can be set to true to indicate that the user completed the course or false if they did not. |
| completedTimestamp | Date | This is the Unix Epoch time that the user completed the course. SAP SuccessFactors Learning uses this completion date for the audit trail of users' completion. |
| grade | String | This is the grade in the course. We do not validate this grade for completion status or rating scales. It is any string that describes the grade that the user earned as the OCN provider sets the grades. |

## **Notifications**

**Introduction**

Integration jobs provide notifications about conditions of the integration process. It will keep all teams and clients are up to date about the integration process. Sending of this notification can be configured for a different schedule and reasons.

**Source of notification**

1. Email notification - Email sent to the list of configured email addresses.

**Notification Reasons**

|  |  |
| --- | --- |
| **Reasons** | **Description** |
| SUCCESS [Admin] | System generated mail send to respected persons while new client configuration has been added successfully. |
| SUCCESS [Batch Job] | System generated mail send to respected persons after successfully running batch jobs with Job ID, status code and total Imports. |
| FAILED [Admin] | System generated failure mail send to respected persons when failure during token update. Error message and status code also send with email body. |
| FAILED [Batch Job] | System generated failure mail send to respected persons when any batch job failed. Error message and status code also send with email body. |
| Empty user file | No any file in FTP folder for inviting users, then system generated mail send to respected persons with error message and status code. |
| Invalid User Import | List of un-import user details has been send by system generated mail with error messages and status code. |
| FTP server is down | System generated mail send while file not transferred during FTP server is down. |
| Coursera Business API down | System generated mail send to respected persons while any of Coursera API is down. |
| LMS API down | System generated mail send to respected persons while any of LMS API is down. |
| Token Exceeds Limit | System generated mail send to respected persons while time limit is exceeds for generating Coursera token. |
| Coursera Refresh Token Not Generated | System generated mail send to respected persons while time limit is exceeds for generating Coursera refresh token. |

## **Limitations**

## **Summary**

The goal of this document is to mention technical limitations pertaining to Coursera and LMS integration for all the Jobs/ Processes i.e.

1. Course Sync Job

2. Course Completion Status Sync Job

3. Invite Users Sync Job

## **I. Following are the Limitation in Job processing at the client side:-**

|  |  |  |  |
| --- | --- | --- | --- |
| Features with Limitations | Course Completion Status sync job | Course Sync Job | Invite Users Sync Job |
| Chunk processing count limit | Yes.  Job reads 250 course status records from Coursera Business API. And post status in a chunk of 250 records at a time to LMS. This limit to fetch record from coursera is based on tested response time for the job to have feasible job completion time | Yes.  Course sync Job reads 250 courses at a time from C4B Apis. This job posts 1000 records at a time to LMS. This limit to fetch record from coursera is based on tested response time for the job to have feasible job completion time | Yes.  This job fetches 100 programs from C4B APIs at a time. In the next step, for each user from the User file received from Successfactor LMS, this job post invitation request against each program. |
| Restartability for job failures | If the job fails it will be re-launched by the AWS batch scheduler to process batch job from the first step itself | If the job fails it will be re-launched by the AWS batch scheduler to process batch job from the first step itself | If the job fails it will be re-launched by the AWS batch scheduler to process batch job from the first step itself |
| Step flow dependency | Yes.  First step involves fetching course completion status from C4B to DB. Second step involves reading status from DB and Publish it to LMS. | Yes.  First step involves reading 250 records at a time from C4B Content API and persist it in a chunk of 1000 records into DB.  Second step reads 1000 records from DB and publish all at one to LMS Course API | Yes.  First step consists of fetching Programs from C4B API in chunks of 100 records at a time and persisted to the DB.  In second step For each user in the file retrieved from LMS FTP server the job reads all the programs and call C4B API for sending user invitation per program. |
| Skip ability | NA | NA | Yes.  In case of Invite Users while Job reads User records from the file retrieved from FTP server at LMS side, if parsing and mapping user records from file to Java results into error due to any reason e.g mandatory field missing, column missing etc in the file. Job will skip that record from processing.  Total skip count of records for the Job is 20 User records after this if any further records come erroneous the complete job will fail. |

## **II. Following are the limitations at Coursera for Business (C4B) API integration with the client application:-**

|  |  |
| --- | --- |
| 1 | All the business APIs are OAuth protected. So to access data from the courser business APIs the request should be accompanied with an oauth token to authenticate request. As a part of one time activity through Admin panel client need to authorize itself against Coursera OAuth server. Integration client redirects user to OAuth Code URL to authorize itself (**with code as grant type**) against the Coursera OAuth server and get the intermediate auth code. |
| 2 | Again as a part of one time activity through Admin panel, this auth code in next step implicitly calls Coursera OAuth server Token URL to receive OAuth token in response along with refresh token. **Auth token expiration time is 1800 seconds**. |
| 3 | During Batch Job execution like Invite or Course sync Job, the client make a call to Coursera Business API to fetch the data along with auth token to get the request authenticated. If the auth token gets expired in between the job execution, the client call to C4B Apis will be failed and that call will be retried with new auth token. In this case client will again implicitly send request using token url with refresh token to regenerate OAuth token and get new auth token in response. If the two subsequent calls to regenerate token requesting time happens within less than 15min then auth server responds with invalid-client-ratelimit error. If the same code is used to generate tokens again invalid-client-ratelimit will occur. |
| 4 | All Coursera business APIs provides pagination variable to fetch data in chunks. It provides start and limit query parameters to get the records in pagination. In client we have configured to fetch 100 records at a time for jobs involved less data processing like fetching programs from Coursera API to invite users to and maximum 250 records at a time for a job that required large data processing like course sync job at a time, because the Coursera API response time increases in proportion to the page size. |

## **III. Following are the limitations at the LMS integration with the Client Application:-**

|  |  |
| --- | --- |
| 1 | All the Api calls to LMS are OAuth protected. So to post data to the SuccessFactor LMS using their APIs, the request should be accompanied with an oauth token to authenticate request. This auth token is generated using LMS auth token URL with client credentials as grant type. During the Post call if the token gets expired then the Post data to LMS gets Unauthorized response. In this case the post data call will fail for that iteration and will be retried with new token. New token will be requested using token URL and refresh token. |
| 2. | In case of course sync job , Maximum 1000 records are configured to be posted to LMS Post course Apis as the response time increases with the number of records to be posted |
| 3. | Minimum time delay required between subsequent calls to LMS APIs is 5 minutes. If it is less than 5 Min LMS responds with server error with response code of the range 5XX and error message as API call limit exceeded. |