Developer Platform (January 2026)

COURSES (COURSE OFFERINGS, TEMPLATES, SCHEMAS)

« Calendar (Events and scheduling) · [home · reference · community | search · index · routing table · scopes table] · Course competencies and objectives »

With the courses resource you can retrieve information about existing courses, modify or remove them, and create new courses.

Contents

Attributes

Note that frequently the course-related JSON data blocks include a *Path* property – these contain a path to the course content published in the service's content store.

Course.**BasicOrgUnit**

This composite contains basic information about an organizational unit to which a course offering is related.

```
{
    "Identifier": <string:D2LID>,
    "Name": <string>,
    "Code": <string>
}
```

Course.**CourseOffering**

The service's fundamental information block for course offerings. Notice that the some fields in this block include **BasicOrgUnit** blocks for the related org units.

```
{
    "Identifier": <string:D2LID>,
    "Name": <string>,
    "Code": <string>,
    "IsActive": <boolean>,
    "Path": <string>,
    "StartDate": <string:UTCDateTime>|null,
    "EndDate": <string:UTCDateTime>|null,
    "LocaleId": <number:D2LID>|null,  // Added with LMS v20.25.8
    "ForceLocale": <boolean>,  // Added with LMS v20.25.8
    "CourseTemplate": { <composite:Course.BasicOrgUnit> }|null,
    "Semester": { <composite:Course.BasicOrgUnit> }|null,
    "Department": { <composite:Course.BasicOrgUnit> }|null,
    "Description": { <composite:RichText> },
    "CanSelfRegister": <boolean>
}
```

ForceLocale

Determines if the course should override the user's saved locale preference.

Course.**CourseOfferingInfo**

When you use actions that update course information for a course offering, you should provide one of these blocks:

```
{
    "Name": <string>,
    "Code": <string>,
    "StartDate": <string:UTCDateTime>|null,
    "EndDate": <string:UTCDateTime>|null,
    "LocaleId": <number:D2LID>|null, // Added with LP API v1.54
    "ForceLocale": <boolean>, // Added with LP API v1.54
    "ShowAddressBook": <boolean>, // Added with LP API v1.54
    "IsActive": <boolean>,
    "Description": { <composite:RichTextInput> },
    "CanSelfRegister": <boolean>|null
}
```

Code

Note that course offering code values have these limitations:

They cannot be more than 50 characters in length.

They may not contain these special characters (or a newline character):

```
\ : * ? " < > | ' # , % &
```

StartDate, *EndDate*

If you provide *null* for these date properties then the back-end service will not overwrite any date already set for the course in the service.

LocaleId

Sets the locale preference of the course. If you provide a value other than *null*, the back-end service updates the locale preference with your identified locale. If you provide *null*, the back-end service preserves whatever locale preference that was already set. This property is required as of LP API v1.54 and forward, but not supported on versions prior to v1.54.

ForceLocale

Determines if the course should override the user’s saved locale preference. If false (the default), the user’s preference can override the course’s value; otherwise, the course’s set locale will not be overridden. This property is required as of LP API v1.54 and forward, but not supported on versions prior to v1.54.

ShowAddressBook

Determines if the email tool’s address book groups together the users enrolled in the course together. If false (the default) users will not be grouped; otherwise, users will be grouped in the address book. This property is required as of LP API v1.54 and forward, but not supported on versions prior to v1.54.

Description

Note that this property is optional to include on input for LP API versions 1.25 and earlier; however, if you include it, the contents are ignored. Note also that this property uses the **RichTextInput** structure type.

CanSelfRegister

Whether or not a user can self-register this course. Null is treated as false. This is required as of API version of LP v1.27, but not supported on versions prior to v1.27.

Course.**CreateCourseOffering**

Block used to create a course.

```
{
    "Name": <string>,
    "Code": <string>,
    "Path": <string>,
    "CourseTemplateId": <number:D2LID>,
    "SemesterId": <number:D2LID>|null,
    "StartDate": <string:UTCDateTime>|null,
    "EndDate": <string:UTCDateTime>|null,
    "LocaleId": <number:D2LID>|null,
    "ForceLocale": <boolean>,
    "ShowAddressBook": <boolean>,
    "Description": { <composite:RichTextInput> },
    "CanSelfRegister": <boolean>|null
}
```

Code

Note that org unit code values have these limitations:

They cannot be more than 50 characters in length.

They may not contain these special characters (or a newline character):

```
\ : * ? " < > | ' # , % &
```

Description

Note that this property is optional to include on input for LP API versions 1.25 and earlier; however, if you include it, the contents are ignored. Note also that this property uses the **RichTextInput** structure type.

ForceLocale

Determines if the course should override the user's saved locale preference.

Path

The root path to use for this course offering's course content. Note that, if your back-end service has path enforcement set on for new org units, then you should leave this property as an empty string, and the back-end service can populate it for you.

SemesterId

Note that if CreateCourse form does not include the Semester element, then you should provide null for this property when creating a course.

ShowAddressBook

Determines if the email tool's address book groups together the users enrolled in the course together.

CanSelfRegister

Whether or not a user can self-register this course. Null is treated as false. This is required as of API version of LP v1.28, but not supported on versions prior to v1.28.

Course.**CourseParent**

```
{
    "CourseOfferingId": <number:D2LID>,
    "Semester": { <composite:Course.BasicOrgUnit> }|null,
    "Department": { <composite:Course.BasicOrgUnit> }|null
}
```

Course.CourseParents

```
{
    "CourseOfferingId": <number:D2LID>,
    "Semesters": null|[ { <composite:Course.BasicOrgUnit> }, ... ],
    "Departments": null|[ { <composite:Course.BasicOrgUnit> }, ... ]
}
```

Course Templates

Course.CourseTemplate

```
{
    "Identifier": <string:D2LID>,
    "Code": <string>,
    "Name": <string>,
    "Path": <string>
}
```

Course.CourseTemplateInfo

```
{
    "Name": <string>,
    "Code": <string>
}
```

Code

Note that org unit code values have these limitations:

They cannot be more than 50 characters in length.
They may not contain these special characters (or a newline character):

```
\ : * ? " < > | ' # , % &
```

Course.CreateCourseTemplate

```
{
    "Name": <string>,
    "Code": <string>,
    "Path": <string>,
    "ParentOrgUnitIds": [ <number:D2LID>, ... ]
}
```

Code

Note that course template code values have these limitations:

They cannot be more than 50 characters in length.
They may not contain these special characters (or a newline character):

```
\ : * ? " < > | ' # , % &
```

Path

The root path to use for this course template's course content. Note that, if your back-end service has path enforcement set on for new org units, then you should leave this property as an empty string, and the back-end service can populate it for you.

Course.SchemaElement

This block describes constraints for parent org units of course offerings (that is, whether particular org units are not permitted as parents for course offerings, or are required as parents for course offerings).

```
{
    "Type": { <composite:OrgUnit.OrgUnitTypeInfo> },
    "IsNotAllowed": <boolean>,
    "IsRequired": <boolean>
}
```

Type

OrgUnit.OrgUnitTypeInfo composite block to describe the org unit type for a course schema.

IsNotAllowed

If true, the org unit type cannot be a parent to the org unit.

IsRequired

If true, the org unit type is required as a parent to the org unit. This value takes precedence over the *IsNotAllowed* property's value.

Source Courses

Course.ReofferCourse

```
{
   "OrgUnitId": <number:D2LID>,
   "CourseCode": <string>,
   "CourseName": <string>,
   "DateCreated": <string:UTCDateTime>,
   "EndDate": <string:UTCDateTime>|null,
   "StartDate": <string:UTCDateTime>|null,
   "IsActive": <boolean>,
   "Deleted": <boolean>
}
```

Course.ReofferedCoursesResult

```
{
   "ReofferedCourses": [ { <composite:Course.ReofferCourse> }, ... ],
}
```

Course.DeploySourceCourseParams

A parameter object for deploying a source course to multiple target course offerings.

```
{
    "TargetCourseOfferingIds": [ <number:D2LID>, ... ]
}
```

TargetCourseOfferingIds

An array of course offering org unit IDs to which the source course will be deployed.

Course.**DeploySourceCourseResult**

The result object for a source course deployment operation with failures.

```
{
    "FailedOrgUnitsWithInfoStr": <string>,
    "FailedOrgUnitsIds": [ <string:D2LID>, ... ],
    "SourceCourseDeployId": <number:D2LID>|null
}
```

FailedOrgUnitsWithInfoStr

String with info about failed org units.

FailedOrgUnitsIds

Array of org unit IDs (as strings) that failed.

SourceCourseDeployId

The deployment ID if one was successfully created.

Course Files

FILESYSTEMOBJECTTYPE_T

The file system object type:

File System Object Type	Value
Folder	1
File	2

Course.**CreateFolder**

```
{
     "RelativePath": <string>
}
```

Course.**CreatedFolder**

```
{
     "RelativePath": <string>,
     "FolderName": <string>,
     "LastModifiedBy" : <number:D2LID>,
     "LastModifiedDate" : <string:UTCDateTime>
}
```

Course.**FileSystemObject**

```
{
    "Name": <string>,
    "FileSystemObjectType": <FILESYSTEMOBJECTTYPE_T>
}
```

Course.**UploadedFile**

```
{
    "RelativePath": <string>,
    "FileName": <string>,
    "Size": <integer>,
    "LastModifiedBy" : <number:D2LID>,
    "LastModifiedDate" : <string:UTCDateTime>
}
```

Copying courses

COURSECOMPONENT_T

Course offering copy jobs can selectively copy distinct course components. These string labels identify the various course components:

- *AttendanceRegisters*
- *Awards*
- *Checklists*
- *Competencies*
- *CompletionTracking*
- *Content*
- *CourseAppearance*
- *CourseFiles*
- *Discussions*
- *DisplaySettings*
- *Dropbox*
- *Faq*
- *Forms*
- *Glossary*
- *Grades*
- *GradesSettings*
- *Groups*
- *Homepages*
- *IntelligentAgents*
- *LearningOutcomes*
- *Links*
- *LtiLink*
- *LtiTP*
- *Navbars*
- *News*
- *QuestionLibrary*
- *Quizzes*
- *ReleaseConditions*
- *Rubrics*
- *S3Model*
- *Schedule*
- *SelfAssessments*
- *Surveys*
- *ToolNames*
- *Widgets*

Note: The *Forms* component label indicates registration forms.

COPYJOBSTATUS_T

Course offering copy jobs will have one of several distinct states, depending on where they are in the conversion process. These string labels describe the states:

Job state	Description
PENDING	Waiting to be picked up for processing.
PROCESSING	Currently in process.
COMPLETE	Processing finished (may contain processing errors: see Conversion history page for details).
FAILED	Processing halted before finish, owing to errors.
CANCELLED	Job was cancelled before finish.

Course.**CreateCopyJobRequest**

When you make a request for a new course offering copy job, you should provide a JSON structure that looks like this one:

```
{
    "SourceOrgUnitId": <number:D2LID>,
    "Components": [ <string:COURSECOMPONENT_T>, ... ]|null,
    "CallbackUrl": <string>|null,
    "DaysToOffsetDates": <number:integer>|null,  // Optional, see notes
    "HoursToOffsetDates": <number:decimal>|null,  // Optional, see notes
    "OffsetByStartDateDifference": <boolean>|null  // Optional, see notes
}
```

SourceOrgUnitId

Must identify an org unit of type Course Offering or Course Template.

Components

JSON array of **course component type names**, for example `"Components": [ "Dropbox", "Grades", "Rubrics" ]`. If this array property is present, *only* the components listed will get copied; if the property is missing or null, then *all* components will get copied.

CallbackUrl

If you provide a valid URL for this property, when the back-end service completes its copy job, it will issue a POST with the provided URL (including any query parameters you provided) with a **`CopyJobComplete`** JSON block in the posted body.

The back-end service expects that your service will respond to that POST with a **200 OK**, and respond in a timely fashion (by default, five seconds or less).

DaysToOffsetDates

If you provide a number (max: 5000) here, all the dates copied into your destination course (e.g. due dates, start dates, and end dates on activities such as quizzes) will be offset by the provided number of days. Enter a positive integer to offset forward, and a negative integer to offset backwards.

You cannot use this parameter at the same time as the *OffsetByStartDateDifference* parameter; doing so will cause a **400 Bad Request** error.

HoursToOffsetDates

If you provide a decimal number (less than 24) here, all the dates copied into your destination course (e.g. due dates, start dates, and end dates on activities such as quizzes) will be offset by the provided number of hours. Enter a positive number to offset forward, and negative to offset backwards.

You cannot use this parameter at the same time as the *OffsetByStartDateDifference* parameter; doing so will cause a **400 Bad Request** error.

OffsetByStartDateDifference

When true, the service will find the difference between the start date of the source course (the course that is being copied from) and the start date of the destination course (the course that is being copied to), and offset all the dates copied into the destination course by that difference:

If the start date of the destination course is earlier than the start date of the source course, the dates will be offset backwards.

If the start date of the destination course is later than the start date of the source course, the dates will be offset forward.

For example, copying from a course with a course start date of July 1, 2018, at 9:00am into a course with a start date of July 3, 2018, at 8:00am will offset all dates 2 days forward and 1 hour backwards.

You cannot use this parameter at the same time as the *DaysToOffsetDates* or *HoursToOffsetDates* parameters; doing so will cause a **400 Bad Request** error.

Course.CreateCopyJobResponse

When the back-end service creates a new job to copy course offerings, it passes you back this JSON structure containing a job token you can use going forward to check on the status of the ongoing job.

```
{
    "JobToken": <string>
}
```

Course.GetCopyJobResponse

When you check on the status of an ongoing job to copy course offerings, the back-end service will pass you back a JSON structure like this one containing a status string:

```
{
    "Status": <string:COPYJOBSTATUS_T>
}
```

Course.CopyJobComplete

When you specify a *CallbackUrl* property in your original request, the back-end service will POST a structure like this to your callback URL:

```
{
    "JobToken": <string>,
    "SourceOrgUnitId": <number:D2LID>,
    "TargetOrgUnitId": <number:D2LID>,
    "Status": <string:COPYJOBSTATUS_T>
}
```

Status

This property will contain either *CANCELLED*, *COMPLETE*, or *FAILED*.

Course.CopyCourseLogMessage

```
{
    "LogId": <number>,
    "CopyCourseJobId": <number:D2LID>,
    "LogDateTime": <string:UTCDateTime>,
    "Message": <string>
}
```

Importing courses

COI_IMPORTJOBSTATUS_T

Course offering import jobs will have one of several distinct states, depending on where they are in the import process. These string labels describe the states:

Job state	Description
UPLOADING	Submission of import job to adaptation process.
PROCESSING	Currently in adaptation process for importing.
PROCESSED	Finished adaptation process.
IMPORTING	Currently importing.
IMPORTFAILED	Import failure.
COMPLETED	Import job completed successfully.

COI_IMPORTLOGMESSAGE_T

Course offering import job log messages can come in a variety of general forms. We use the term *COI_IMPORTLOGMESSAGE_T* to stand in for an appropriate identifying integer value.

Import log message type	Value
System	0
Progress	1
Warning	2
Error	3

COI_IMPORTOPERATION_T

Course offering import jobs can be created for a number of underlying import operations. We use the term *COI_IMPORTOPERATIONTYPE_T* to stand in for an appropriate identifying integer value.

Import operation type	Value
Conversion	1
Import	2

Course.GetImportJobResponse

When you check on the status of an ongoing job to import a course offering, the back-end service will pass you back a JSON structure like this one containing a status string.

```
{
    "JobToken": <string>,
    "TargetOrgUnitId": <number:D2LID>,
    "Status": <string:COI_IMPORTJOBTSTATUS_T>
}
```

When you specify a *CallbackUrl* parameter on your original import request, the back-end service will post a structure like this to your callback URL, once, when the import job completes (either successfully or unsuccessfully).

Course.ImportCourseLog

```
{
    "LogId": <number:D2LID>,
    "ConversionImportJobId": <number:D2LID>,
    "OperationTypeId": <string:COI_IMPORTOPERATION_T>,
    "LogDateTime": <string:UTCDateTime>,
    "Message": <string>,
    "TypeId": <string:COI_IMPORTLOGMESSAGE_T>
}
```

Actions

`DELETE /d2l/api/lp/(version)/courses/(orgUnitId)`

Delete a course offering.

Parameters: **version** (**D2LVERSION**) – API version.
orgUnitId (**D2LID**) – Org unit ID.

Oauth2 Scopes: *orgunits:course:delete*

Status Codes: **200 OK** – Action succeeded.
403 Forbidden – No permission to manage or delete courses.
404 Not Found – Org unit not found or not a course offering.
429 Too Many Requests – API **call-rate limit** exceeded.

API Versions: **1.49+** – Route first appears in LMS v20.25.1.
1.46-1.48 – **Deprecated** as of LMS v20.26.1.
1.45- – **Obsolete** as of LMS v20.26.1.

> **Note**
>
> Using this action is equivalent to using the route to **`send an org unit to the recycle bin`**. You can **`restore the course offering from the recycle bin`**, if needed.

`GET /d2l/api/lp/(version)/courses/schema`

Retrieve the list of parent org unit type constraints for course offerings.

Parameters: **version** (**D2LVERSION**) – API version.

Status Codes: **200 OK** – Action succeeded.
429 Too Many Requests – API **call-rate limit** exceeded.

API Versions: **1.49+** – Route first appears in LMS v20.25.1.
1.46-1.48 – **Deprecated** as of LMS v20.26.1.
1.45- – **Obsolete** as of LMS v20.26.1.

Return. This action returns a JSON array of **`SchemaElement`** blocks.

`GET /d2l/api/lp/(version)/courses/(orgUnitId)`

Retrieve a course offering.

Parameters: **version** (**D2LVERSION**) – API version.
orgUnitId (**D2LID**) – Org unit ID.

Oauth2 Scopes: *orgunits:course:read*

Status Codes:
- **200 OK** – Action succeeded.
- **403 Forbidden** – No permission to see course offerings.
- **404 Not Found** – Org unit not found or not a course offering.
- **429 Too Many Requests** – API **call-rate limit** exceeded.

API Versions:
- **1.49+** – Route first appears in LMS v20.25.1.
- **1.46-1.48** – **Deprecated** as of LMS v20.26.1.
- **1.45-** – **Obsolete** as of LMS v20.26.1.

Return. This action returns a **`CourseOffering`** JSON block with the provided course's information.

`GET /d2l/api/lp/(version)/courses/(orgUnitId)/image`

Retrieve the course image for a course offering.

Parameters:
- **version** (**D2LVERSION**) – API version.
- **orgUnitId** (**D2LID**) – Org unit ID.

Query Parameters:
- **width** (*integer*) – Optional. Desired width (pixels) of course image.
- **height** (*integer*) – Optional. Desired height (pixels) of course image.

Oauth2 Scopes: *organizations:image:read*

Status Codes:
- **200 OK** – Action succeeded.
- **403 Forbidden** – No permission to see course offerings.
- **404 Not Found** – Org unit not found or not a course offering.
- **429 Too Many Requests** – API **call-rate limit** exceeded.

API Versions:
- **1.49+** – Route first appears in LMS v20.25.1.
- **1.46-1.48** – **Deprecated** as of LMS v20.26.1.
- **1.45-** – **Obsolete** as of LMS v20.26.1.

Return. This action returns a file stream containing the current course offering's image.

You can use the query parameters to prompt the service to re-size the image to a useful dimension. You may want to perform ad-hoc testing against your back-end service to determine the best sizes to ask for: the service will send you back an image that's at least as large as the dimensions you provide.

Currently, the default size for the course offering image is 300 pixels in height and width.

`GET /d2l/api/lp/(version)/courses/parents`

Retrieve parent departments and semesters for the provided courses.

Parameters:
- **version** (**D2LVERSION**) – API version.

Query Parameters:
- **orgUnitIdsCSV** (**CSV** of **D2LIDs**) – List of org unit IDs (limited to 25 or fewer).

Oauth2 Scopes: *enrollment:own_enrollment:read*

Status Codes:
- **200 OK** – Action succeeded.
- **403 Forbidden** – orgUnitIdsCSV parameter is empty or contains org units the user is not enrolled in.
- **404 Not Found** – No enrollments or no permission to call this route.
- **429 Too Many Requests** – API **call-rate limit** exceeded.

API Versions:
- **1.55+** – Route first appears in LMS v20.25.12.

Input. When calling this action, you must provide a list of org unit IDs for course offerings in which the calling user is actively enrolled.

Return. This action returns a JSON array of **`CourseParent`** blocks.

> **Note**
>
> Note that as of LMS v20.25.12, this action supersedes the **`/d2l/api/lp/(version)/courses/parentorgunits`** action: you should use this action by preference.

GET /d2l/api/lp/(*version*)/courses/parentorgunits

Retrieve parent department and semester for the provided courses.

Parameters: **version** (**D2LVERSION**) – API version.

Query Parameters: **orgUnitIdsCSV** (**CSV** of **D2LIDs**) – List of org unit IDs (limited to 25 or fewer).

Oauth2 Scopes: *enrollment:own_enrollment:read*

Status Codes:
- **200 OK** – Action succeeded.
- **403 Forbidden** – orgUnitIdsCSV parameter is empty or contains org units the user is not enrolled in.
- **404 Not Found** – No enrollments or no permission to call this route.
- **429 Too Many Requests** – API **call-rate limit** exceeded.

API Versions:
- **1.49+** – Route first appears in LMS v20.25.1.
- **1.46-1.48** – **Deprecated** as of LMS v20.26.1.
- **1.45-** – **Obsolete** as of LMS v20.26.1.

Input. When calling this action, you must provide a list of org unit IDs for course offerings in which the calling user is actively enrolled.

Return. This action returns a JSON array of **`CourseParent`** blocks.

> **Note**
>
> As this action retrieves references to only a single parent department and semester for each course, it is no longer preferred as of LMS v20.25.12. Instead, use the **`/d2l/api/lp/(version)/courses/parents`** action.

POST /d2l/api/lp/(*version*)/courses/

Create a new course offering.

Parameters: **version** (**D2LVERSION**) – API version.

JSON Parameters: **CreateCourseOffering** (**`Course.CreateCourseOffering`**) – New course offering properties.

Oauth2 Scopes: *orgunits:course:create*

Status Codes:
- **200 OK** – Action succeeded.
- **400 Bad Request** – Invalid course data provided, or could not create new course.

403 Forbidden – No permission to manage or create courses.

404 Not Found – Could not resolve org unit for created course to a known org unit.

429 Too Many Requests – API **call-rate limit** exceeded.

API Versions:

1.49+ – Route first appears in LMS v20.25.1.

1.46-1.48 – **Deprecated** as of LMS v20.26.1.

1.45- – **Obsolete** as of LMS v20.26.1.

Return. This action returns a **`CourseOffering`** JSON block for the newly created course.

> **Note**
>
> CanSelfRegister field is required as of API version of LP v1.28, but not supported on versions prior to v1.28.

PUT /d2l/api/lp/(*version*)/courses/(*orgUnitId*)

Update a current course offering.

Parameters:

version (**D2LVERSION**) – API version.

orgUnitId (**D2LID**) – Org unit ID.

JSON Parameters:

CourseOfferingInfo (**`Course.CourseOfferingInfo`**) – Updated course offering properties.

Oauth2 Scopes: *orgunits:course:update*

Status Codes:

200 OK – Action succeeded.

400 Bad Request – Invalid course data provided.

403 Forbidden – No permission to update courses.

404 Not Found – Org unit not found or not a course offering.

429 Too Many Requests – API **call-rate limit** exceeded.

API Versions:

1.49+ – Route first appears in LMS v20.25.1.

1.46-1.48 – **Deprecated** as of LMS v20.26.1.

1.45- – **Obsolete** as of LMS v20.26.1.

Input. This action replaces the associated course offering’s data with all the properties you provide.

> **Note**
>
> CanSelfRegister field is required as of API version of LP v1.27, but not supported on versions prior to v1.27.

PUT /d2l/api/lp/(*version*)/courses/(*orgUnitId*)/image

Update the course image for a course offering.

Parameters:

version (**D2LVERSION**) – API version.

orgUnitId (**D2LID**) – Org unit ID.

Oauth2 Scopes: *organizations:image:update*

Status Codes:

200 OK – Action succeeded.

403 Forbidden – No permission to see course offerings.

404 Not Found – Org unit not found or not a course offering.
429 Too Many Requests – API **call-rate limit** exceeded.

API Versions:
1.49+ – Route first appears in LMS v20.25.1.
1.46-1.48 – **Deprecated** as of LMS v20.26.1.
1.45- – **Obsolete** as of LMS v20.26.1.

Input. Via **simple upload**, provide an image file with these restrictions:

- The content disposition *name* property for the file part in the upload should be “Image”, but the name case is not sensitive.
- The file must be less than 2000 KB in size.
- The file must be of a type commonly and natively renderable in a web browser (for example, JPEG, GIF, or PNG).

Course Templates

Course templates provide a way to gather sets of default properties for more efficient course offering creation.

`DELETE /d2l/api/lp/(version)/coursetemplates/(orgUnitId)`

Delete a course template.

Parameters:
version (**D2LVERSION**) – API version.
orgUnitId (**D2LID**) – Org unit ID.

Status Codes:
200 OK – Action succeeded.
403 Forbidden – No permission to manage or delete courses.
404 Not Found – No such org unit, or not a course template.
429 Too Many Requests – API **call-rate limit** exceeded.

API Versions:
1.49+ – Route first appears in LMS v20.25.1.
1.46-1.48 – **Deprecated** as of LMS v20.26.1.
1.45- – **Obsolete** as of LMS v20.26.1.

> **Note**
>
> Using this action is equivalent to using the route to `send an org unit to the recycle bin`. You can `restore the course template from the recycle bin`, if needed.

`GET /d2l/api/lp/(version)/coursetemplates/(orgUnitId)`

Retrieve a course template.

Parameters:
version (**D2LVERSION**) – API version.
orgUnitId (**D2LID**) – Org unit ID.

Status Codes:
200 OK – Action succeeded.
403 Forbidden – No permission to see course info.
404 Not Found – No such org unit, or not a course template.
429 Too Many Requests – API **call-rate limit** exceeded.

API Versions:
1.49+ – Route first appears in LMS v20.25.1.

1.46-1.48 – **Deprecated** as of LMS v20.26.1.
1.45- – **Obsolete** as of LMS v20.26.1.

Return. This action returns a `CourseTemplate` JSON block.

`GET /d2l/api/lp/(version)/coursetemplates/schema`

Retrieve the list of parent org unit type constraints for course offerings built on this template.

Parameters: **version** (**D2LVERSION**) – API version.

Status Codes:
200 OK – Action succeeded.
429 Too Many Requests – API **call-rate limit** exceeded.

API Versions:
1.49+ – Route first appears in LMS v20.25.1.
1.46-1.48 – **Deprecated** as of LMS v20.26.1.
1.45- – **Obsolete** as of LMS v20.26.1.

Return. This action returns a JSON array of `SchemaElement` blocks.

`POST /d2l/api/lp/(version)/coursetemplates/`

Create a new course template.

Parameters: **version** (**D2LVERSION**) – API version.

JSON Parameters: **CreateCourseTemplate** (`Course.CreateCourseTemplate`) – New course template properties.

Oauth2 Scopes: *orgunits:coursetemplate:create*

Status Codes:
200 OK – Action succeeded.
400 Bad Request – Invalid course template data provided, or could not create new course template.
403 Forbidden – No permission to manage or create courses and templates.
404 Not Found – Could not resolve org unit for created course template to a known org unit.
429 Too Many Requests – API **call-rate limit** exceeded.

API Versions:
1.49+ – Route first appears in LMS v20.25.1.
1.46-1.48 – **Deprecated** as of LMS v20.26.1.
1.45- – **Obsolete** as of LMS v20.26.1.

Return. This action returns a `CourseTemplate` JSON block containing the data for the new course template.

`PUT /d2l/api/lp/(version)/coursetemplates/(orgUnitId)`

Update the information for a course template.

Parameters:
version (**D2LVERSION**) – API version.
orgUnitId (**D2LID**) – Org unit ID.

JSON Parameters: **CourseTemplateInfo** (`Course.CourseTemplateInfo`) – Updated course template properties.

Status Codes:
200 OK – Action succeeded.
400 Bad Request – Invalid course template data provided.
403 Forbidden – No permission to manage courses.

404 Not Found – No such org unit, or not a course template.

429 Too Many Requests – API **call-rate limit** exceeded.

API Versions:

1.49+ – Route first appears in LMS v20.25.1.

1.46-1.48 – **Deprecated** as of LMS v20.26.1.

1.45- – **Obsolete** as of LMS v20.26.1.

Input. This action uses the data in your provided block to completely replace the associated course template's data on the service.

Source Courses

Brightspace source courses allow for re-offering and deployment of course-offerings based on a defined source of materials.

`GET /d2l/api/lp/(version)/sourceCourses/(orgUnitId)/currentReofferedCourse`

Retrieve the current course offering for a given source course.

Parameters:

version (**D2LVERSION**) – API version.

orgUnitId (**D2LID**) – Org unit ID.

Query Parameters:

subOrganizationOrgUnitId (**D2LID**) – Optional (see following description). Filter to retrieve only the current offering under the sub-organization org unit.

Oauth2 Scopes: *orgunits:sourcecourses:read*

Status Codes:

200 OK – Action succeeded.

403 Forbidden – No permission to See Source Courses.

404 Not Found – Unable to find current source course re-offering or unable to find the source course.

429 Too Many Requests – API **call-rate limit** exceeded.

API Versions:

1.53+ – Route first appears in LMS v20.25.10.

Input. If you do not provide the *subOrganizationOrgUnitId* query parameter, the action retrieves the current offering under the Organization (if found), given the calling user has permission to see source courses at the organization level. If you do provide this query paramter, this action retrieves only a course offering under the provided sub-organization org unit (if any), given the calling user has permission to see source courses at that sub-organization org unit level.

Return. This action returns a `SourceCourses` JSON block containing the current course offering for the provided source course ID.

`GET /d2l/api/lp/(version)/sourceCourses/(orgUnitId)/reofferedCourses`

Retrieve the re-offered courses for a given source course.

Parameters:

version (**D2LVERSION**) – API version.

orgUnitId (**D2LID**) – Org unit ID.

Query Parameters:

subOrganizationOrgUnitId (**D2LID**) – Optional (see following description). Filter to retrieve only re-offered courses under the sub-organization org unit.

Oauth2 Scopes: *orgunits:sourcecourses:read*

Status Codes:
- **200 OK** – Action succeeded.
- **403 Forbidden** – No permission to See Source Courses.
- **404 Not Found** – Unable to find specified source course.
- **429 Too Many Requests** – API **call-rate limit** exceeded.

API Versions:
- **1.53+** – Route first appears in LMS v20.25.10.

Input. If you do not provide the *subOrganizationOrgUnitId* query parameter, the action retrieves the re-offered courses under the Organization (if found), given the calling user has permission to see source courses at the organization level. If you do provide this query parameter, this action retrieves only course offerings under the provided sub-organization org unit (if any), given the calling user has permission to see source courses at that sub-organization org unit level.

Return. This action returns a `SourceCourses` JSON block containing the re-offered courses for the provided source course ID.

`POST /d2l/api/lp/(version)/sourceCourses/(orgUnitId)/deploy`

Deploy a source course to one or more target course offerings.

Parameters:
- **version** (**D2LVERSION**) – API version.
- **orgUnitId** (**D2LID**) – Source course org unit ID.

JSON Parameters:
- **DeploySourceCourseParams** (`DeploySourceCourseParams`) – Deployment parameters.

Oauth2 Scopes: *manageCourses:deploy:manage*

Status Codes:
- **200 OK** – Action succeeded, all deployments to targets initiated.
- **207 Multi Status** – Partial success, some deployments failed. See response for details.
- **400 Bad Request** – Bad request (missing/invalid parameters, invalid targets, or other problem).
- **401 Unauthorized** – Not authorized (org unit not in org, missing permissions, or user not enrolled in source course).
- **403 Forbidden** – Feature disabled.
- **404 Not Found** – Source course or target not found.
- **429 Too Many Requests** – API **call-rate limit** exceeded.

API Versions:
- **1.53+** – Route first appears in LMS v20.25.10.

Input: If you provide the *subOrganizationOrgUnitId* query parameter, the calling user may only use this action with org units that are descendants of the provided sub-organization org unit, given the calling user has permission to see source courses and deploy source courses at the sub-organization org unit level.

Note that target course offerings are reset before deploying the source course contents to them.

Return: On partial success, this action returns a `DeploySourceCourseResult` JSON block. On full success, returns a deployment ID (number) identifying the deployment action. If the back-end service does not return a success status code, it did not initiate a source course deployment.

Course files

`DELETE /d2l/api/lp/(version)/(orgUnitId)/managefiles/file`

Delete a course file.

Parameters: **version** (**D2LVERSION**) – API version.
orgUnitId (**D2LID**) – Org unit ID.

JSON Parameters: **DeleteFile** – Delete file properties.

Query Parameters: **path** (*string*) – Path to the file to delete relative to the course path.

Oauth2 Scopes: *managefiles:files:manage*

Status Codes:
- **200 OK** – Action succeeded.
- **400 Bad Request** – Invalid input data provided.
- **403 Forbidden** – No permission to See the File Listing or Delete Files.
- **404 Not Found** – Org Unit not found

API Versions:
- **1.49+** – Route first appears in LMS v20.25.1.
- **1.46-1.48** – **Deprecated** as of LMS v20.26.1.
- **1.45-** – **Obsolete** as of LMS v20.26.1

`DELETE /d2l/api/lp/(version)/(orgUnitId)/managefiles/folder`

Delete a course folder.

Parameters: **version** (**D2LVERSION**) – API version.

JSON Parameters: **DeleteFolder** – Delete folder properties.

Query Parameters: **path** (*string*) – Path to the folder to delete relative to the course path.

Oauth2 Scopes: *managefiles:folders:manage*

Status Codes:
- **200 OK** – Action succeeded.
- **400 Bad Request** – Invalid input data provided.
- **403 Forbidden** – No permission to See the File Listing or Delete Files.
- **404 Not Found** – Org Unit not found

API Versions:
- **1.49+** – Route first appears in LMS v20.25.1.
- **1.46-1.48** – **Deprecated** as of LMS v20.26.1.
- **1.45-** – **Obsolete** as of LMS v20.26.1

> **Note**
>
> Delete folder is fully recursive; the back-end service will also delete all sub-folders and files contained within the identified folder.

`GET /d2l/api/lp/(version)/(orgUnitId)/managefiles/`

Retrieve the direct child contents (folders and files) of an org unit path.

Parameters: **version** (**D2LVERSION**) – API version.

Query Parameters: **path** (*string*) – Optional. Path relative to the course path, if not specified the course path is used.

Oauth2 Scopes: *managefiles:files:read managefiles:folders:read*

Status Codes:
- **200 OK** – Action succeeded.
- **403 Forbidden** – No permission to See the File Listing.
- **404 Not Found** – No such path, or no such org unit.

	429 Too Many Requests – API **call-rate limit** exceeded.
API Versions:	**1.49+** – Route first appears in LMS v20.25.1.

Return. This action returns a **`object list page`** containing the resulting **`FileSystemObject`** blocks for the segment specified. The objects are returned folders first followed by files both sorted alphabetically by name. If there are more segments the Next field will be set with the request required for the next segment.

`GET /d2l/api/lp/(`*`version`*`)/(`*`orgUnitId`*`)/managefiles/file`

Retrieve a course file specified by the path.

Parameters:	**version** (**D2LVERSION**) – API version.
Query Parameters:	**path** (*string*) – Path to the file relative to the course path.
Oauth2 Scopes:	*managefiles:files:read*
Status Codes:	**200 OK** – Action succeeded. **403 Forbidden** – No permission to see the course file. **404 Not Found** – No such course file, or no such org unit. **429 Too Many Requests** – API **call-rate limit** exceeded.
API Versions:	**1.49+** – Route first appears in LMS v20.25.1. **1.46-1.48** – **Deprecated** as of LMS v20.26.1. **1.45-** – **Obsolete** as of LMS v20.26.1.

Input. You must provide the *path* query string paramter. It needs to contain a valid path to a file in the current org unit’s content space relative to the org unit’s root path (typically something like `/topic1/topic1.html`).

Return. This action returns a file stream containing the course file data.

`POST /d2l/api/lp/(`*`version`*`)/(`*`orgUnitId`*`)/managefiles/file/save`

Saves an uploaded course file to specified path.

Parameters:	**version** (**D2LVERSION**) – API version. **orgUnitId** (**D2LID**) – Org unit ID.
Query Parameters:	**overwriteFile** (*boolean*)
Form Parameters:	**fileKey** – File key value identifying the uploaded file. **relativePath** – The path relative to the course path where the file will be uploaded.
Oauth2 Scopes:	*managefiles:files:manage*
Status Codes:	**200 OK** – Action succeeded. **400 Bad Request** – Invalid input data provided. **403 Forbidden** – No permission to See the File Listing or Upload Files. **404 Not Found** – Org Unit not found, or RelativePath does not exist. **409 Conflict** – If the OverwriteFile query parameter is not used or set to false, and a file with the same name already exists in the RelativePath. Conflict - File already exists.
API Versions:	**1.49+** – Route first appears in LMS v20.25.1. **1.46-1.48** – **Deprecated** as of LMS v20.26.1. **1.45-** – **Obsolete** as of LMS v20.26.1

Input. Provide a POST form with the file key the service sent you from your **`resumable file upload`** action. Note that you can provide a name parameter for the framework to use for the file; if you provide no name parameter, the framework seeks to gather the file name from the uploaded file itself.

If you provide a *relativePath* it should describe the folder path between the established course path and the folder you want to contain your uploaded file; for example *temp* or *temp/upload*, or similar.

> **Note**
>
> Before you can use this action to save a course file, the containing folder must exist; if you provide a relative path, then all folders in that relative path must already exist.

Return. This action returns a **`UploadedFile`** JSON block for the newly uploaded file.

`POST /d2l/api/lp/(`*`version`*`)/(`*`orgUnitId`*`)/managefiles/file/upload`

Initiate a resumable file upload request for a course file.

Parameters:
- **version** (**D2LVERSION**) – API version.
- **orgUnitId** (**D2LID**) – Org unit ID.

Oauth2 Scopes: *managefiles:files:manage*

Status Codes:
- **308 Resume Incomplete** – Request to start a resumable upload successfully received and initiated, with response indicating upload path and file key.
- **400 Bad Request** – Invalid input data provided.
- **403 Forbidden** – No permission to See the File Listing or Upload Files.
- **404 Not Found** – Org Unit not found.

API Versions:
- **1.49+** – Route first appears in LMS v20.25.1.
- **1.46-1.48** – **Deprecated** as of LMS v20.26.1.
- **1.45-** – **Obsolete** as of LMS v20.26.1

Input. Initiate a **resumable upload** process. Your POST request must contain three special headers:

X-Upload-Content-Type to indicate the mime type of the file you're uploading.

X-Upload-Content-Length to indicate the total size of the file you're uploading, in bytes (not the size of the part passed in any individual request).

X-Upload-File-Name to indicate the file name for the file you're uploading.

Return. If successful, this action initiates a resumable upload process. As part of that process, the service provides you a file upload path you can use to make further requests to upload parts of your file, and a file key value that you can use to **`save the uploaded file`** to the course once you have uploaded the entire content length of your file.

`POST /d2l/api/lp/(`*`version`*`)/(`*`orgUnitId`*`)/managefiles/folder`

Create a folder in course.

Parameters:
- **version** (**D2LVERSION**) – API version.

orgUnitId (**D2LID**) – Org unit ID.

JSON Parameters: **CreateFolder** (**`Course.CreateFolder`**) – Create folder properties.

Oauth2 Scopes: *managefiles:folders:manage*

Status Codes:

- **200 OK** – Action succeeded.
- **400 Bad Request** – Invalid input data provided.
- **403 Forbidden** – No permission to See the File Listing or Create and Edit Files.
- **404 Not Found** – Org Unit not found.

API Versions:

- **1.49+** – Route first appears in LMS v20.25.1.
- **1.46-1.48** – **Deprecated** as of LMS v20.26.1.
- **1.45-** – **Obsolete** as of LMS v20.26.1

Return. This action returns a **`CreatedFolder`** JSON block for the newly created folder.

Copying courses

`GET /d2l/api/le/(version)/import/(orgUnitId)/copy/(jobToken)`

Retrieve the status of a queued course copy job request.

Parameters:

- **version** (**D2LVERSION**) – API version.
- **orgUnitId** (**D2LID**) – Org unit ID of the target course offering.
- **jobToken** (*string*) – Course copy job token.

Status Codes:

- **200 OK** – Action succeeded.
- **403 Forbidden** – No permission to retrieve status of copy job.
- **404 Not Found** – No such target org unit, no such source org unit (from copy job), no such copy job, or job token is invalid.
- **429 Too Many Requests** – API **call-rate limit** exceeded.

API Versions:

- **1.82+** – Route first appears in LMS v20.25.1.
- **1.75-81** – **Deprecated** as of LMS v20.26.1..
- **1.74-** – **Obsolete** as of LMS v20.26.1.

Return. This action returns a **`GetCopyJobResponse`** JSON block.

`POST /d2l/api/le/(version)/import/(orgUnitId)/copy/`

Queue up a new course copy job request.

Parameters:

- **version** (**D2LVERSION**) – API version.
- **orgUnitId** (**D2LID**) – Org unit ID of the target course offering.

JSON Parameters: **CreateCopyJobRequest** (**`Course.CreateCopyJobRequest`**) – Job request for course copy.

Status Codes:

- **202 Accepted** – Job request accepted.
- **400 Bad Request** – Source org unit not found, invalid course component string value, invalid (empty) course component list, or invalid combination of date offset parameters.
- **403 Forbidden** – No permission to create course copy job request.
- **404 Not Found** – Target org unit not found.
- **429 Too Many Requests** – API **call-rate limit** exceeded.

API Versions:

- **1.82+** – Route first appears in LMS v20.25.1.

1.75-81 – **Deprecated** as of LMS v20.26.1..
1.74- – **Obsolete** as of LMS v20.26.1.

This action requests the creation of a new course offering copy job to copy one or more course offering components from a source org unit to the target org unit identified in the route path.

> **Note**
>
> The source org unit you provide must be of type Course Offering or Course Template.
>
> The target org unit you provide in the path *must* exist before you create a copy job.
>
> The job processor in the back-end service will do no check to verify that your request course components have *already* been copied from the source to the target course.

Return. This action returns a **`CreateCopyJobResponse`** JSON block.

GET /d2l/api/le/(*version*)/ccb/logs

Retrieve the list of logs for course copy jobs.

Parameters:

version (**D2LVERSION**) – API version.

Query Parameters:

bookmark (*string*) – Optional. Bookmark to use for fetching next data set segment.

pageSize (*number*) – Optional. Number of log entries (maximum) to return in each data page (default is *20*, maximum is *100*).

sourceOrgUnitId (*number*) – Optional. Org unit ID from which components where copied.

destinationOrgUnitId (*number*) – Optional. Org unit ID to which components where copied.

subOrganizationOrgUnitId (**D2LID**) – Optional (see following description). Enables access to logs for org units under sub-organization org unit.

startDate – Optional. Fetch logs of course copy jobs started since *startDate*.

endDate – Optional. Fetch logs of course copy jobs started before *endDate*.

Status Codes:

200 OK – Action succeeded.

400 Bad Request – Provided endDate is early than startDate.

403 Forbidden – No permission to retrieve logs of copy course jobs.

404 Not Found – No copy course job were found.

429 Too Many Requests – API **call-rate limit** exceeded.

API Versions:

1.91+ – Route first appears in LMS v20.26.1.

Input. You can use the optional *subOrganizationOrgUnitId* parameter to constrain this action's effect to a sub-org, with these constraints: (i) the calling user must have permission to view copy-course component logs at the level of the indicated sub-org's *type*, and (ii) both the source and destination org unit for this action must be *descendants* of the indicated sub-organization.

Return. This action returns an **`object list page`** containing the resulting **`CopyCourseLogMessage`** data blocks for the segment following your *bookmark* parameter (or the first segment if the parameter is empty or missing).

Importing courses

`GET /d2l/api/le/(version)/import/(orgUnitId)/imports/(jobToken)`

Retrieve the status of a queued course import job request.

Parameters:
- **version** (**D2LVERSION**) – API version.
- **orgUnitId** (**D2LID**) – Org unit ID.
- **jobToken** (*string*) – Course copy job token.

Oauth2 Scopes: *import:job:fetch*

Status Codes:
- **200 OK** – Action succeeded.
- **403 Forbidden** – No permission to retrieve status of import job.
- **404 Not Found** – No such target org unit, no such source org unit (from import job), no such import job, or job token is invalid.
- **429 Too Many Requests** – API **call-rate limit** exceeded.

API Versions:
- **1.82+** – Route first appears in LMS v20.25.1.
- **1.75-81** – **Deprecated** as of LMS v20.26.1..
- **1.74-** – **Obsolete** as of LMS v20.26.1.

Return. This action returns a **`GetImportJobResponse`** JSON block.

`GET /d2l/api/le/(version)/import/(orgUnitid)/imports/(jobToken)/logs/`

Retrieve the logs for a course import job.

Parameters:
- **version** (**D2LVERSION**) – API version.
- **orgUnitId** (**D2LID**) – Org unit ID.
- **jobToken** (*string*) – Course copy job token.

Query Parameters:
- **bookmark** (*string*) – Optional. Bookmark to use for fetching next data set segment.

Status Codes:
- **200 OK** – Action succeeded.
- **404 Not Found** – No logs found for course import job.
- **429 Too Many Requests** – API **call-rate limit** exceeded.

API Versions:
- **1.82+** – Route first appears in LMS v20.25.1.
- **1.75-81** – **Deprecated** as of LMS v20.26.1..
- **1.74-** – **Obsolete** as of LMS v20.26.1.

Return. This action returns a **`paged result set`** containing the resulting **`ImportCourseLog`** blocks for the segment following your *bookmark* parameter (or the first segment if that parameter is empty or missing).

`POST /d2l/api/le/(version)/import/(orgUnitId)/imports/`

Create a new course import job request.

Parameters:
- **version** (**D2LVERSION**) – API version.

orgUnitId (**D2LID**) – Org unit ID of the target course offering.

Query Parameters: **callbackUrl** (*string*) – Callback URL to post status to.

Oauth2 Scopes: *import:job:create*

Status Codes: **202 Accepted** – Job request accepted.

403 Forbidden – No permission to create course import job request.

404 Not Found – No such org unit.

429 Too Many Requests – API **call-rate limit** exceeded.

API Versions: **1.82+** – Route first appears in LMS v20.25.1.

1.75-81 – **Deprecated** as of LMS v20.26.1..

1.74- – **Obsolete** as of LMS v20.26.1.

Input. Provide the course package to import using the **simple file upload** process: provide the course package as the upload data to the POST form; set the *filename* field in the *Content-Disposition* part-header to the base name of the file you are uploading; set the *name* field in the part-header to *file*:

```
Content-Disposition: form-data; name="file"; filename="ImportPackage.zip"
Content-Type: application/zip
```

The service will create a new course import job for your course package, and return a job token you can use to **`query for the job's status`**.

If you provide a *callbackUrl* query parameter, it should be a URL that the back-end service can reach, and that you can receive. If you do provide such an URL, then the back-end service will post a **`GetImportJobStatusResponse`** JSON block to that URL when the import job completes, successfully or unsuccessfully.

Return. This action returns a simple JSON block containing a token for the import job you can use later to check on the job's status:

```
{
   "JobToken": <string>
}
```

« Calendar (Events and scheduling) · [home · reference · community | search · index · routing table · scopes table] · Course competencies and objectives »