

Developer Platform (January 2026)

COURSE CONTENT (CONTENT OBJECTS, TABLES OF

CONTENT)

« Grades (grade objects, categories, values, schemes) · [home · reference · community | search · index · routing table
· scopes table] · Course updates for end users »

Attributes

CONTENT_T

Course content come in two types: course modules and course topics; we use the term CONTENT_T to stand in for an appropriate integer value:

Content object type	Value
<i>Module</i>	0
<i>Topic</i>	1

ACTIVITYTYPE_T

Course objects are grouped into a set of more granular activity types and we use the integer values to stand for the appropriate activity types:

Content activity type	Value
<i>UnknownActivity</i>	-1
<i>Module</i>	0
<i>File</i>	1
<i>Link</i>	2
<i>Dropbox</i>	3
<i>Quiz</i>	4
<i>DiscussionForum</i>	5
<i>DiscussionTopic</i>	6
<i>LTI</i>	7 // Legacy LTI (v1.1)
<i>Chat</i>	8 // Deprecated
<i>Schedule</i>	9
<i>Checklist</i>	10
<i>SelfAssessment</i>	11
<i>Survey</i>	12
<i>OnlineRoom</i>	13 // Deprecated
<i>CourseLink</i>	14
<i>Scorm_1_3</i>	20
<i>Scorm_1_3_Root</i>	21
<i>Scorm_1_2</i>	22
<i>Scorm_1_2_Root</i>	23
<i>Scorm</i>	24 // Content Service SCORM
<i>Lor</i>	25
<i>LorScorm</i>	26
<i>LTIAdvantage</i>	27 // LTI version v1.3+
<i>OrgUnit</i>	28
<i>ActivityInstance</i>	29

TOPIC_T

Contents

Attributes

Content overview
Scheduled items
Table of contents
User Progress
Pacing
Captions

Actions

Content overview
Scheduled items
Table of contents
User Progress
Pacing
Captions
Order

Course topic objects are grouped into a set of types and we use the term `TOPIC_T` to stand in for an appropriate integer value.

Content topic type	Value
<i>File</i>	1
<i>Link</i>	3
<i>SCORM_2004</i>	5 ^
<i>SCORM_2004_ROOT</i>	6 ^
<i>SCORM_1_2</i>	7 ^
<i>SCORM_1_2_ROOT</i>	8 ^

^ Reserved for future implementation

COMPLETION_T

The back-end service lets you filter content items based on their completion status, and we use a numeric value to indicate the filtering policy to follow:

Completion status	Value
<i>Any</i>	1
<i>CompletedOnly</i>	2
<i>NotCompletedOnly</i>	3

CONTENT_COMPLETIONTYPE_T

Completion type can be set on a content topic depending on how the topic should be completed. We use the term `CONTENT_COMPLETIONTYPE_T` to stand in for an appropriate integer value. Exempt topics will always return 3, signifying the completion type *None*.

Completion type	Value
<i>Manual</i>	1
<i>Auto</i>	2
<i>None</i>	3

Manual, Auto

These two completion types indicate that the content topic is “required”.

None

This completion type indicates that the content topic is exempt, or “optional”.

CONTENTCOMPLETIONLEVEL_T

Completed and required topic counts within an org unit are rolled up at one of the following levels.

Completion level type	Value
<i>OrgUnit</i>	1
<i>RootModule</i>	2
<i>Topic</i>	3

EXEMPTION_T

The back-end service lets you filter content items based on their exemption status, and we use a numeric value to indicate the filtering policy to follow:

Exemption status	Value
<i>Any</i>	1
<i>ExemptedOnly</i>	2
<i>NotExemptedOnly</i>	3

Content.**ContentObject**

When you use an action to retrieve course content structure information, you will receive one of two types of ContentObject JSON blocks (which use the *Type* property to indicate which type it is):

IsLocked

Some SCORM content packages can support sequencing, locking the topic ordering. Content objects with this property set *True* don't permit content re-ordering or navigating the content out of sequence.

Type

When you use an action that retrieves information from the service for a course's content, you get back one of two types of ContentObject JSON blocks (as indicated in the ContentObject's *Type* property). In modules, this property has the value 0; in topics, this property has the value 1.

Module.

```
{
  "Structure": [ <Content.ContentObject>, ... ],
  "ModuleStartDate": <string:UTCDateTime>|null,
  "ModuleEndDate": <string:UTCDateTime>|null,
  "ModuleDueDate": <string:UTCDateTime>|null,
  "IsHidden": <boolean>,
  "IsLocked": <boolean>,
  "Id": <number:D2LID>,
  "Title": <string>,
  "ShortTitle": <string>,
  "Color": <string>|null, // Added with LMS 20.24.7
  "Type": 0,
  "Description": { <composite:RichText> }|null,
  "ParentModuleId": <number:D2LID>|null,
  "Duration": <number>|null,
  "LastModifiedDate": <string:UTCDateTime>|null
}
```

ModuleDueDate

If this property is present and not null, the content module has a due date. If the property is present with a non-null value, it contains a **UTCDateTime**.

ParentModuleId

If null, the module is a root module and the "parent" is the owning org unit; otherwise, this property contains the ID of the parent content module.

Title

The module title as shown on the dashboard page in the LMS.

Color

The hexadecimal color value for the module, null if unset.

Duration

Minutes of class time that are likely needed to cover the material in class.

Topic.

Note that when a topic refers to an object, the properties in the retrieved topic are those in the underlying referred-to object. For example, the topic's *title* is taken from the underlying object's title.

```
{
  "TopicType": <number:TOPIC_T>,
  "Url": <string>|null,
  "StartDate": <string:UTCDateTime>|null,
  "EndDate": <string:UTCDateTime>|null,
  "DueDate": <string:UTCDateTime>|null,
  "IsHidden": <boolean>,
  "IsLocked": <boolean>,
  "IsBroken": <boolean>, // Added with LE API 1.72
  "OpenAsExternalResource": <boolean>|null,
  "Id": <number:D2LID>,
  "Title": <string>,
  "ShortTitle": <string>,
  "Type": 1,
}
```

```

"Description": { <composite:RichText> }|null,
"ParentModuleId": <number:D2LID>,
"ActivityId": <string>|null,
"Duration": <number>|null, // Available in LE's unstable contract
"IsExempt": <boolean>,
"ToolId": <number:D2LID>|null,
"ToolItemId": <number:D2LID>|null,
"ActivityType": <number:ACTIVITYTYPE_T>,
"GradeItemId": <number:D2LID>|null,
"LastModifiedDate": <string:UTCDateTime>|null,
"AssociatedGradeItemIds": [<number:D2LID>, ...]
}

```

DueDate

If this property is present and not null, the content topic has a due date. If the property is present with a non-null value, it contains a **UTCDateTime**.

IsBroken

If true, the content topic is a broken link (the file it refers to has been deleted or the associated item no longer exists).

OpenAsExternalResource

In link-type topics (*TopicType* value 3), this property set to *True* will open the link as an external resource in a new window. If the property is set to *False*, the link will open in the current window. If the property is null, or not present, the underlying tool determines the default behaviour. For topic types other than link-type topics, this property will always have a value of null.

ParentModuleId

Contains the ID of the parent content module.

Title

The topic title as shown on the viewContent page in the LMS.

ActivityId

If not null, identifies the underlying activity for this content topic.

Duration

Minutes of class time that are likely needed to cover the material in class. Note that this field is not available except in the *unstable* contract version.

IsExempt

As of LE v1.25, Activities can be exempt from being graded, and if the content topic's activity has been exempted, then the corresponding Content Topic will return its exemption status.

ToolId

If not null, identifies the underlying activity's tool id for this content topic.

ToolItemId

If not null, identifies the underlying activity's id within it's tool domain.

ActivityType

Identifies the **ACTIVITYTYPE_T** of this topic, or the underlying activity's type.

GradeItemId

If not null, identifies the underlying activity's grade item id.

AssociatedGradeItemIds

Identifies all associated grade item ids (including the value found in *GradeItemId*). Not all underlying activities will support multiple associated grade items. In the case of no associated grade items then this list will be empty.

Content.ContentObjectData

When you use an action to provide course content information, you use one of two types of `ContentObjectData` JSON blocks (and use the `Type` property to indicate which type it is):

Description

This property is optional and may be empty or null. Note also that it uses the **RichTextInput** structure type.

ShortTitle

This field may be empty or null.

Title

This field may not be empty, null, or consist only of white-space characters.

Type

In modules, this property has the value 0; in topics, this property has the value 1.

Url

In link-type topics (*TopicType* value 3), this property should be the URL you want to fetch when the user opens the link-type topic.

In file-type topics (*TopicType* value 1), this property should contain a new path that is valid within the course offering's content space and indicate where the system should place the file, for example:

```
/content/enforced/8083-EXT-101/test_topic_file.pdf
```

Module.

```
{
  "Title": <string>,
  "ShortTitle": <string>,
  "Type": 0,
  "ModuleStartDate": <string:UTCDateTime>|null,
  "ModuleEndDate": <string:UTCDateTime>|null,
  "ModuleDueDate": <string:UTCDateTime>|null,
  "IsHidden": <boolean>,
  "IsLocked": <boolean>,
  "Description": { <composite:RichTextInput> }|null,
  "Duration": <number>|null // Available in LE's unstable contract
}
```

ModuleStartDate, ModuleEndDate

If you provide *null* for one of these fields, you will erase any prior value set for the field.

ModuleDueDate

As of LE API v1.70, if you provide *null* for this field, you will erase any prior value set for the field (with older versions of the API, providing *null* would leave a prior value unaffected).

Note that if the *ModuleDueDate* field is present and has a non-null value, the content module has a due date; otherwise, it does not.

Duration

Minutes of class time that are likely needed to cover the material in class.

Topic.

```
{
  "Title": <string>,
  "ShortTitle": <string>,
  "Type": 1,
  "TopicType": <number:TOPIC_T>,
  "Url": <string>,
  "StartDate": <string:UTCDateTime>|null,
  "EndDate": <string:UTCDateTime>|null,
  "DueDate": <string:UTCDateTime>|null,
  "IsHidden": <boolean>,
  "IsLocked": <boolean>,
  "OpenAsExternalResource": <boolean>|null,
}
```

```

"Description": { <composite:RichTextInput> }|null,
"MajorUpdate": <boolean>|null,
"MajorUpdateText": <string>|null,
"ResetCompletionTracking": <boolean>|null,
"Duration": <number>|null // Available in LE's unstable contract
}

```

StartDate, EndDate, DueDate

As of LE API v1.70, if you provide *null* for one of these fields, you will erase any prior value set for the field (with older versions of the API, providing *null* would leave prior values unaffected).

Note that if the *DueDate* field is present and has a non-null value, the content topic has a due date; otherwise, it does not.

MajorUpdate

When a topic is being updated, this property set to *True* will cause students to be notified of the change. This property otherwise has a value of null.

MajorUpdateText

When a topic is being updated and *MajorUpdate* is set to *True*, this value will be the notification summary. This property otherwise has a value of null.

OpenAsExternalResource

In link-type topics (*TopicType* value 3), this property set to *True* will open the link as an external resource in a new window. If the property is set to *False*, the link will open in the current window. If the property is null, or not present, the underlying tool determines the default behaviour. For topic types other than link-type topics, this property will always have a value of null.

ResetCompletionTracking

When a topic is being updated and *MajorUpdate* is set to *True*, this property set to *True* will cause the topic's completion tracking to be reset. This property otherwise has a value of null.

Duration

Minutes of class time that are likely needed to cover the material in class.

Note

When a topic refers to an object, the properties in the retrieved topic are those in the underlying referred-to object. For example, the topic's *title* is taken from the underlying object's title.

Exemptions.**ItemExemptionInfo**

```

{
  "ExemptItems": [ // Array of D2LID numbers that are content object IDs
    <number:D2LID>,
    <number:D2LID>,
    <number:D2LID>, ...
  ]
}

```

Exemptions.**ItemExemptionInfoFilter**

```

{
  "ExemptStartDate": <string:UTCDateTime>|null,
  "ExemptEndDate": <string:UTCDateTime>|null,
  "ExcludeGradedActivities": <boolean>|null,
  "ExcludeCompleteActivities": <boolean>|null
}

```

Content overview

Each course offering org unit can have a content overview consisting of a **rich text description** and an optional attachment.

Overview.**Overview**

```
{
  "Description": <composite:RichText>,
  "HasAttachment": <boolean>
}
```

Scheduled items

Content.**ScheduledItem**

A scheduled item describes an activity in an org unit that was added to the content tool with start date, end date, or due date.

```
{
  "UserId": <string:D2LID>,
  "OrgUnitId": <string:D2LID>,
  "ItemId": <number:D2LID>,
  "ItemName": <string>,
  "ItemType": <number:CONTENT_T>,
  "ItemUrl": <string>|null,
  "StartDate": <string:UTCDateTime>|null,
  "EndDate": <string:UTCDateTime>|null,
  "DueDate": <string:UTCDateTime>|null,
  "CompletionType": <number:CONTENT_COMPLETIONTYPE_T>,
  "DateCompleted": <string:UTCDateTime>|null,
  "ActivityType": <number:ACTIVITYTYPE_T>,
  "IsExempt": <boolean>
}
```

DueDate

If this property is present and not null, and *IsExempt* is *false*, then the content topic has a due date.

CompletionType

Completion type of the content topic. This value is set to 3 (completion type *None*) when the item is exempt or optional.

IsExempt

If true, the user is exempt from the topic; otherwise, false. This property is only present if item exemptions are supported.

An activity is considered *scheduled* for a particular user when it is accessible to that user:

- The user is enrolled in the org unit with a role that can have content assigned to them

- The item is a content topic in the org unit's content

- The item is accessible to the user, considering a variety of factors:

- Start and end dates of the org unit
- Scheduled dates for the item itself
- Publication status of the item
- Any special access status for the item
- Applicable release conditions

Content.**ScheduledItemCount**

The count of content items with a start, end, or due date, for the provided org unit context.

```
{
  "OrgUnitId": <string:D2LID>,
  "UserId": <string:D2LID>,
  "ItemCount": <number>
}
```

Content.**OverdueItem**

An overdue item describes an activity in an org unit that was added to the content tool with a due date and that is now overdue for a particular user.

```
{
  "UserId": <number:D2LID>,
  "OrgUnitId": <number:D2LID>,
  "ItemId": <number:D2LID>,
  "ItemName": <string>,
  "DueDate": <string:UTCDateTime>
}
```

An activity is considered *overdue* for a particular user when these considerations are met:

- The item is a content topic in the org unit's content
- The item is accessible to the user, considering a variety of factors:
 - Start and end dates of the org unit
 - Scheduled dates for the item itself
 - The item has a due date scheduled *and* that date is in the past
 - The user hasn't yet completed the item
 - The item has not been exempted for the user

Table of contents

ToC.TableOfContents

This is a nested structure. The *TableOfContent* comprises a list of *Module* blocks. Each *Module* block can contain a list of *Topic* blocks, and can contain a list of child *Module* blocks. Thus, each module can have a child that's a topic, or another module, but topics are "leaf nodes".

```
{
  "Modules": [ // Array of Module blocks
    {
      "ModuleId": <number:D2LID>,
      "Title": <string>,
      "Modules": [ // Array of child Module blocks
        {
          "ModuleId": <number:D2LID>,
          "Title": <string>,
          "SortOrder": <number>,
          "StartDateTime": <string:UTCDateTime>|null,
          "EndDateTime": <string:UTCDateTime>|null,
          "Modules": [ { <composite:ToC.Module> }, ... ],
          "Topics": [ { <composite:ToC.Topic> }, ... ],
          "IsHidden": <boolean>,
          "IsLocked": <boolean>,
          "PacingStartDate": <string:ISODate>|null,
          "PacingEndDate": <string:ISODate>|null,
          "DefaultPath": <string>,
          "LastModifiedDate": <string:UTCDateTime>|null
        },
        { <composite:Module> }, ... ],
      "Topics": [ // Array of Topic blocks
        {
          "TopicId": <number:D2LID>,
          "Identifier": <string:D2LID>,
          "TypeIdentifier": <string>,
          "Title": <string>,
          "Bookmarked": <boolean>,
          "Unread": <boolean>,
          "Url": <string>,
          "SortOrder": <number>,
          "StartDateTime": <string:UTCDateTime>|null,
          "EndDateTime": <string:UTCDateTime>|null,
          "ActivityId": <string>|null,
          "CompletionType": <number:CONTENT_COMPLETIONTYPE_T>,
          "IsExempt": <boolean>,
          "IsHidden": <boolean>,
          "IsLocked": <boolean>,
          "IsBroken": <boolean>,
          "ToolId": <number:D2LID>|null,
          "ToolItemId": <number:D2LID>|null,
          "ActivityType": <number:ACTIVITYTYPE_T>,
          "GradeItemId": <number:D2LID>|null,
          "LastModifiedDate": <string:UTCDateTime>|null,

```



```

    "StartDateAvailabilityType": <AVAILABILITY_T>|null, // Available in LE's unstable
    "EndDateAvailabilityType": <AVAILABILITY_T>|null // Available in LE's unstable con
  },
  { <composite:Topic> }, ... ]
},
{ <composite:Module> }, ... ]
}

```

ModuleId

Contains a **D2LID** for the module.

TopicId

Contains a **D2LID** for the topic. Note that this property contains the same identifier as the *Identifier* property, but formatted as a JSON number rather than as a string.

Url

Carries the same value as appears in a **ContentObjectData** block: for link content topics, it's the content link; for file content topics, it's the path in the course content space for the file data.

SortOrder

A non-negative integer describing the intended, ascending, order of the topics within a given module (or of modules within a given module).

ActivityId

Carries the same value as appears in a **ContentObject** block.

CompletionType

Completion type of the content topic.

IsExempt

If true, the user is exempt from the topic; otherwise, false.

IsLocked

Carries the same value as appears in a **ContentObject** block.

PacingStartDate

If this property is present and not null, the content module has a pacing start date. If the property is present with a non-null value, it contains a **ISODate**.

```
<string:ISODate>|null
```

PacingEndDate

If this property is present and not null, the content module has a pacing end date. If the property is present with a non-null value, it contains a **ISODate**.

```
<string:ISODate>|null
```

IsBroken

If true, the activity for the topic can't be found; otherwise, false.

DefaultPath

The suggested default file path for topics within this module.

ToolId

If not null, identifies the underlying activity's tool id for this content topic.

ToolItemId

If not null, identifies the underlying activity's id within its tool domain.

ActivityType

Identifies the **ACTIVITYTYPE_T** of this topic, or the underlying activity's type.

GradeItemId

If not null, identifies the underlying activity's grade item id.

StartDateAvailabilityType, EndDateAvailabilityType

Identifies the availability type (**AVAILABILITY_T**) for the start and end date values for the content object.

User Progress

ContentCompletions.**ContentTopicCompletion**

When the client uses an action to retrieve a content topic's completion,

```
{
  "UserId": <number:D2LID>,
  "OrgUnitId": <number:D2LID>,
  "TopicId": <number:D2LID>,
  "CompletionType": <number:CONTENT_COMPLETIONTYPE_T>,
  "CompletionDate": <string:UTCDateTime>|null
}
```

UserId

User the content topic completion belongs to.

OrgUnitId

OrgUnit of the content topic.

TopicId

ID of the content topic.

CompletionType

Completion type of the content topic.

CompletionDate

Date content topic was completed, or null if the topic has not been completed.

ContentCompletions.**ContentTopicCompletionUpdate**

When the client uses an action to update a content topic's completion,

```
{
  "CompletedDate": <string:UTCDateTime>|null
}
```

CompletedDate

Date content topic was completed, or null to indicate that the topic has not been completed.

ContentCompletions.**ContentAggregateCompletion**

When the client uses an action to retrieve content aggregate completion,

```
{
  "UserId": <string:D2LID>,
  "OrgUnitId": <string:D2LID>,
  "ObjectId": <number:D2LID>,
  "Title": <string>,
  "RequiredItems": <number>,
  "CompletedItems": <number>
}
```

UserId

ID of the user for whom the completion data is reported.

OrgUnitId

ID of the org unit that the content object belongs to.

ObjectId

ID of the content object for which the required child topics are counted. If data is aggregated at OrgUnit level, the value is 0.

Title

Content object title. If data is aggregated at OrgUnit level, the value is the org unit name.

RequiredItems

This count includes completed topics which meet all of the following criteria: visible to the user, not marked as exempt for the user, and are required topics.

CompletedItems

This count includes all *RequiredItems* plus completed topics not visible to the user and any completed optional topics.

ContentCompletions.ContentLearnerProgress

When the client uses an action to retrieve content learner progress,

```
{
  "UserId": <string:D2LID>,
  "OrgUnitId": <string:D2LID>,
  "RequiredItems": <number>,
  "CompletedItems": <number>
}
```

UserId

ID of the user for whom the completion data is reported.

OrgUnitId

ID of the org unit that the content object belongs to.

RequiredItems

This count includes completed topics which meet all of the following criteria: visible to the user, not marked as exempt for the user, and are required topics.

CompletedItems

This count includes all *RequiredItems* plus completed topics not visible to the user and any completed optional topics.

UserProgress.UserProgressData

When the client uses an action to retrieve the user's progress through content, the back-end service reports on the progress using blocks that look like this:

```
{
  "ObjectId": <number:D2LID>,
  "UserId": <number:D2LID>,
  "CompletedDate": <string:UTCDateTime>|null,
  "IsRead": <boolean>,
  "NumVisits": <number>,
  "TotalTime": <number>,
  "LastVisited": <string:UTCDateTime>|null,
  "Visited": <boolean>,
  "Completed": <boolean>,
  "IsExempt": <boolean>
}
```

ObjectId

ID of the content topic in question.

TotalTime

Total time spent on the page in seconds. You cannot update this property to a lower value than its current value; also, the service will not update the value to a time past the last login time of the user.

LastVisited

Date and time the user last visited the content.

Visited

If true, the user has visited the page; otherwise, false.

IsRead

If true, the user has read the page; otherwise, false.

IsExempt

If true, the user is exempt from the activity; otherwise, false.

Pacing

Pacing.**PacingInfo**

```
{
  "StartDate": <string:ISODate>|null,
  "EndDate": <string:ISODate>|null
}
```

Pacing.**ModulePacingData**

```
{
  "StartDate": <string:ISODate>|null,
  "EndDate": <string:ISODate>|null
}
```

Pacing.**PacingData**

```
{
  "StartDate": <string:ISODate>|null,
  "EndDate": <string:ISODate>|null,
  "SkipWeekends": <boolean>
}
```

SkipWeekends

If true, dates sent on content modules will not be set on weekends.

Captions

Captions.**CaptionsMetadata**

A captions metadata object describes a captions file. Each captions file is associated with a particular audio or video topic in course content, providing captions data for a specific language.

This object only contains file metadata. It does not contain the actual contents of the captions file.

```
{
  "Filename": <string>,
  "LanguageCode": <string>,
  "LanguageCulture": <string>, // Optional.
  "LanguageName": <string>
}
```

Filename

The name of the captions file, as it appears in Organization Files. Includes file extension.

LanguageCode

The two-letter ISO 639-1 language code of the language this captions file is associated with.

LanguageCulture

The two-letter ISO 639-1 culture code of the language this captions file is associated with. Some languages do not correspond to a specific culture, in which case this property will not be present.

LanguageName

The translated, human-readable name of the language this captions file is associated with. This property is translated to the language used by the LMS.

Actions

DELETE `/d2l/api/le/(version)/(orgUnitId)/content/modules/(moduleId)`

Delete a specific module from an org unit.

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

Oauth2 Scopes: *content:modules:manage*

Status Codes: **200 OK** – Action succeeded.
 400 Bad Request – Content object with provided module ID is not a module.
 403 Forbidden – No permission to delete course content.
 404 Not Found – Module ID not found in provided org unit, or 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.

Note

As of LMS v20.23.4 applications should use the *content:modules:manage* scope to gain access to this action. While the *content:modules:read* scope will still provide access, use of *content:modules:read* for this action is now **deprecated**.

DELETE `/d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)`

Delete a specific topic from an org unit.

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

Oauth2 Scopes: *content:topics:manage*

Status Codes: **200 OK** – Action succeeded.
 400 Bad Request – Content object with provided topic ID is not a topic.
 403 Forbidden – No permission to delete course content.
 404 Not Found – Topic ID not found in provided org unit, or 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.

Note

As of LMS v20.23.4 applications should use the *content:topics:manage* scope to gain access to this action. While the *content:topics:read* scope will still provide access, use of *content:topics:read* for this action is now **deprecated**.

GET /d2l/api/le/(version)/(orgUnitId)/content/modules/(moduleId)

Retrieve a specific module for an org unit.

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

Oauth2 Scopes: *content:modules:readonly*

Status Codes: **200 OK** – Action succeeded.
 400 Bad Request – Content object with provided module ID is not a module.
 403 Forbidden – No permission to view course content.
 404 Not Found – Module ID not found in provided org unit, or 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.

Return. This action returns a **ContentObject** JSON data block of type **Module**.

Note

As of LMS v20.23.4 applications should use the *content:modules:readonly* scope to gain access to this action. While the *content:modules:read* scope will still provide access, use of *content:modules:read* for this action is now **deprecated**.

GET /d2l/api/le/(version)/(orgUnitId)/content/modules/(moduleId)/access/

Retrieve a list of users with access to a specified module.

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

Query Parameters: **userId (D2LID)** – Optional. Retrieve access for a single user.
 roleId (D2LID) – Optional. Retrieve access for users with the given role.

Oauth2 Scopes: *content:access:read*

Status Codes: **200 OK** – Action succeeded.
 400 Bad Request – Content object with provided module ID is not a module.
 403 Forbidden – No permission to view and manage course content, see classlist, view user enrollment.
 404 Not Found – Module ID not found in provided org unit, or 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.

Return. This action returns a **object list page** containing the resulting **UserAccess** blocks for the segment following your *bookmark* parameter (or the first segment if that parameter is empty or missing).

GET /d2l/api/le/(version)/(orgUnitId)/content/modules/(moduleId)/structure/

Retrieve the structure for a specific module in an org unit.

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

Oauth2 Scopes: *content:modules:readonly*

Status Codes: **200 OK** – Action succeeded.
 400 Bad Request – Content object with provided module ID is not a module.
 403 Forbidden – No permission to view course content.

404 Not Found – Module ID not found in provided org unit, or 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.

Return. This action returns a JSON array of **ContentObject** data blocks (can be either **Module** or **Topic** type blocks).

Note

As of LMS v20.23.4 applications should use the *content:modules:readonly* scope to gain access to this action. While the *content:modules:read* scope will still provide access, use of *content:modules:read* for this action is now **deprecated**.

GET `/d2l/api/le/(version)/(orgUnitId)/content/root/`

Retrieve the root module(s) for an org unit.

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

Oauth2 Scopes: *content:modules:readonly*

Status Codes: **200 OK** – Action succeeded.
403 Forbidden – No permission to view course content.
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 JSON array of **ContentObject** data blocks of type **Module**.

Note

As of LMS v20.23.4 applications should use the *content:modules:readonly* scope to gain access to this action. While the *content:modules:read* scope will still provide access, use of *content:modules:read* for this action is now **deprecated**.

GET `/d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)`

Retrieve a specific topic for an org unit.

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

Oauth2 Scopes: *content:topics:readonly*

Status Codes: **200 OK** – Action succeeded.
400 Bad Request – Content object with provided topic ID is not a topic.
403 Forbidden – No permission to view course content.
404 Not Found – Topic ID not found in provided org unit, or 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.

Return. This action returns a **ContentObject** JSON data block of type **Topic**.

Note

As of LMS v20.23.4 applications should use the `content:topics:readonly` scope to gain access to this action. While the `content:topics:read` scope will still provide access, use of `content:topics:read` for this action is now **deprecated**.

GET /d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)/access/

Retrieve a list of users with access to a specified content topic.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID. topicId (D2LID) – Topic ID.
Query Parameters:	userId (D2LID) – Optional. Retrieve access for a single user. roleId (D2LID) – Optional. Retrieve access for users with the given role.
Oauth2 Scopes:	<code>content:access:read</code>
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Content object with provided topic ID is not a topic. 403 Forbidden – No permission to view and manage course content, see classlist, view user enrollment. 404 Not Found – Topic ID not found in provided org unit, or 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.

Return. This action returns a **object list page** containing the resulting **UserAccess** blocks for the segment following your *bookmark* parameter (or the first segment if that parameter is empty or missing).

GET /d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)/file

Retrieve the content topic file for a content topic.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID. topicId (D2LID) – Topic ID.
Query Parameters:	stream (boolean) – Optional. If set to true, will stream the file instead of downloading it.
Oauth2 Scopes:	<code>content:file:read</code>
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Content topic is not a file. 403 Forbidden – No permission to view course content. 404 Not Found – Topic ID not found in provided org unit, or org unit not found, or content topic's file 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.

Return. This action retrieves the underlying file for a file content topic.

You can use the optional *stream* query parameter to change the content disposition of the returned data; if you don't provide the parameter (or set it to a value of false), the returned *Content-Disposition* header indicates the data payload is an "attachment"; otherwise, the header indicates the data payload is "inline".

Note

For LE API versions 1.34 and older, this action requires the `content:topics:read` scope; for LE API versions 1.35 and newer, it instead requires the `content:file:read` scope.

POST /d2l/api/le/(version)/(orgUnitId)/content/exemptions/users/(userId)/byList/

Create exemptions for specific content topics for a user.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID. userId (D2LID) – User for whom to create the exemptions.
JSON Parameters:	ItemExemptionInfo (ItemExemptionInfo) – Exemption info for item exemptions.
Oauth2 Scopes:	<i>content:exemptions:write</i>
Status Codes:	200 OK – Action succeeded. 403 Forbidden – No permission to create exemptions 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.

Returns. This action returns the created exemptions expressed in a **ItemExemptionInfo** JSON block.

POST

/d2l/api/le/(version)/(orgUnitId)/content/exemptions/users/(userId)/byFilter/

Create exemptions for content topics matching a specific filter for a user.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID. userId (D2LID) – User for whom to create the exemptions.
JSON Parameters:	ItemExemptionInfoFilter (ItemExemptionInfoFilter) – Filter to select content items for exemption.
Oauth2 Scopes:	<i>content:exemptions:write</i>
Status Codes:	200 OK – Action succeeded. 403 Forbidden – No permission to create exemptions 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.

Returns. This action returns the created exemptions expressed in a **ItemExemptionInfo** JSON block.

POST

/d2l/api/le/(version)/(orgUnitId)/content/exemptions/users/(userId)/byList/delete/

Delete exemptions for content topics for a user.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID. userId (D2LID) – User for whom to delete the exemptions.
JSON Parameters:	ItemExemptionInfo (ItemExemptionInfo)
Oauth2 Scopes:	<i>content:exemptions:delete</i>
Status Codes:	200 OK – Action succeeded. 403 Forbidden – No permission to delete exemptions 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.

Returns. This action returns the deleted expressions in a **ItemExemptionInfo** JSON block.

POST /d2l/api/le/(version)/(orgUnitId)/content/modules/(moduleId)/structure/

Add a child module or topic to a specific module's structure.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID. moduleId (D2LID) – Module ID for the parent module.
Query Parameters:	base64 (boolean) – Optional. If present and true, when adding a file topic, you are providing base64-encoded file attachment data. renameFileIfExists (boolean) – Optional. If present and true, when adding a file topic, if a file already exists with the same name then it will rename this new one and continue rather than overwrite it.
JSON Parameters:	varies – Varies (see input description).
OAuth2 Scopes:	<i>content:modules:manage</i>
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Provided parent module ID does not identify a module, invalid child topic data provided, or OpenAsExternalResource is set to true on a topic type other than a link. 403 Forbidden – No permission to create or edit course content. 404 Not Found – Parent module ID not found in provided org unit, or org unit not found, or attempt to re-use existing file failed because existing file 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.

Input. You can use this action in several ways, depending on whether you're attempting to add a child module, a child link topic, or a child file topic to an existing (parent) module.

Module. If you're adding a child module to a parent module's structure, you should provide a POST body containing a single part: a **ContentObjectData** JSON data block of type **Module**.

Link Topic. If you're adding a child topic to a parent module's structure that consists of a **Link type topic (3)**, you should provide a POST body containing a single part: a **ContentObjectData** JSON data block of type **Topic**. In this case, the *Url* property in the data block points to the resource you want the link to point to.

File Topic. If you're adding a child topic to a parent module's structure that consists of a **File type topic (1)**, you should provide a multipart/mixed post body with two parts (as with the **simple file upload** process): the first, a **ContentObjectData** JSON data block of type **Topic**; the second, the file attachment data itself that you want to store in the org unit's content area.

When providing a file, the ContentObjectData's *Url* property must contain a path that's valid for the current org unit's content space, identifying the path to the new file, and the name the new file will have in the content space (typically something like */content/enforced/orgunit_id-org_id/newfilename*).

If you want to re-use a file in the org unit's content space, you can provide a ContentObjectData block that points to the existing file, indicate a *File* topic type (1), and not post any attached file data.

Return. On success, this action returns a JSON data block containin the properties for the content object you've created:

If you're adding a module, this action returns a **ContentObject** JSON data block of type **Module**.

If you're adding a file or link type topic, this action returns a **ContentObject** JSON data block of type **Topic**.

Note

As of LMS v20.23.4 applications should use the *content:modules:manage* scope to gain access to this action. While the *content:modules:read* scope will still provide access, use of *content:modules:read* for this action is now **deprecated**.

POST /d2l/api/le/(version)/(orgUnitId)/content/root/

Create a new root module for an org unit.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID.
JSON Parameters:	ContentObjectData (Content.ContentObjectData of type Module) – New root module property data.
Oauth2 Scopes:	<i>content:modules:manage</i>
Status Codes:	200 OK – Action succeeded. 403 Forbidden – No permission to create or edit course content. 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 **ContentObject** JSON data block of type **Module**.

Note

As of LMS v20.23.4 applications should use the *content:modules:manage* scope to gain access to this action. While the *content:modules:read* scope will still provide access, use of *content:modules:read* for this action is now **deprecated**.

PUT /d2l/api/le/(version)/(orgUnitId)/content/modules/(moduleId)

Update a particular module for an org unit.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID. moduleId (D2LID) – Module ID.
JSON Parameters:	ContentObjectData (Content.ContentObjectData of type Module) – Updated module property data.
Oauth2 Scopes:	<i>content:modules:manage</i>
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Invalid module data provided, or module ID provided does not identify a module. 403 Forbidden – No permission to create or edit course content. 404 Not Found – No such module found in org unit, or 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.

Input. Note that you cannot use this action to affect a module's existing Structure property.

Note

As of LMS v20.23.4 applications should use the *content:modules:manage* scope to gain access to this action. While the *content:modules:read* scope will still provide access, use of *content:modules:read* for this action is now **deprecated**.

PUT /d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)

Update a particular topic for an org unit.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID. topicId (D2LID) – Topic ID.
JSON Parameters:	ContentObjectData (Content.ContentObjectData of type Topic) – Updated topic property data.
Oauth2 Scopes:	<i>content:topics:manage</i>

Status Codes:**200 OK** – Action succeeded.**400 Bad Request** – Invalid topic data provided, topic ID provided does not identify a topic, or `OpenAsExternalResource` is set to true on a topic type other than a link.**403 Forbidden** – No permission to create or edit course content.**404 Not Found** – No such topic found in org unit, or 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.

Input. Note that you *can* use this action to change a topic's *Url* property, but if it changes the new URL must be a valid one for the org unit's content store (the service makes no check that the URL actually points at anything useful, however). You *cannot* use this action to provide new file data for an existing topic.

Note

As of LMS v20.23.4 applications should use the `content:topics:manage` scope to gain access to this action. While the `content:topics:read` scope will still provide access, use of `content:topics:read` for this action is now **deprecated**.

PUT /d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)/file

Replace the content topic file for a content topic.

Parameters:**version (D2LVERSION)** – API version.**orgUnitId (D2LID)** – Org unit ID.**topicId (D2LID)** – Topic ID.**Oauth2 Scopes:** `content:file:write`**Status Codes:****200 OK** – Action succeeded.**400 Bad Request** – Invalid file data provided or topic ID provided does not identify a file topic type.**403 Forbidden** – No permission to create or edit course content.**404 Not Found** – No such topic found in org unit, or 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.

Input. Provide an uploaded content file using the **simple file upload** process; the content-disposition part header for the file part should have the name *file*. This file will be saved in the same location as the existing content topic file; note that the existing content topic file will not be removed by this action.

Note

For LE API versions 1.34 and older, this action requires the `content:topics:read` scope; for LE API versions 1.35 and newer, it instead requires the `content:file:write` scope.

Content overview**GET /d2l/api/le/(version)/(orgUnitId)/overview**

Retrieve the content overview for a course offering.

Parameters:**version (D2LVERSION)** – API version.**orgUnitId (D2LID)** – Org unit ID.**Status Codes:****200 OK** – Action succeeded.**403 Forbidden** – No permission to retrieve course offering's content overview.**404 Not Found** – Course has no content overview set.

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 **Overview** JSON data block containing the course offering's content overview details.

GET /d2l/api/le/(version)/(orgUnitId)/overview/attachment

Retrieve the content overview's file attachment for a course offering.

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

Status Codes: 200 OK – Action succeeded.
403 Forbidden – No permission to retrieve course offering's content overview or overview attachment.
404 Not Found – Course has no content overview set, or has no overview attachment, or cannot find attachment file.
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 file stream containing the course offering's content overview file attachment.

Scheduled items

Scheduled items are activities in content that have assigned dates (start, end, and/or due), and become therefore, scheduled for a user's attention. Start and end dates govern the accessibility of the activity, while a due date specifies a date after which the item will become overdue if the user does not *complete* it.

GET /d2l/api/le/(version)/content/myItems/

Retrieve the calling user's scheduled items.

Parameters: version (D2LVERSION) – API version.

Query Parameters: completion (COMPLETION_T) – Optional. Completion status.
orgUnitIdsCSV (CSV of D2LIDs) – List of org unit IDs (limited to 100 or fewer).
startDateTime (UTCDateTime) – Optional. Start of time window to examine.
endDateTime (UTCDateTime) – Optional. End of time window to examine.

Status Codes: 200 OK – Action succeeded.
400 Bad Request – Invalid org unit ID or provided start date later than end date.
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. When calling this action, you must provide a list of org unit IDs that amount to some or all of the calling user's active enrollments. You can also filter the data retrieved based on its **completion status** or whether an item's due date falls within a provided time window. Exempted items do not have due dates.

Return. This action returns an **ObjectListPage** JSON block containing a list of **ScheduledItem** blocks, sorted by the earliest of start date, end date, or due date that match the provided query parameter filters.

This action ignores invalid selection data on input (unknown or inaccessible org units or items), returning only those items that the calling user has access to.

GET /d2l/api/le/(version)/content/items/(userId)

Retrieve the scheduled items for a particular user, within a number of org units.

Parameters:	version (D2LVERSION) – API version. userId (D2LID) – User for which to retrieve the scheduled items for.
Query Parameters:	completion (COMPLETION_T) – Optional. Completion status. orgUnitIdsCSV (CSV of D2LIDs) – List of org unit IDs (limited to 100 or fewer). startDateTime (UTCDateTime) – Optional. Start of time window to examine. endDateTime (UTCDateTime) – Optional. End of time window to examine. exemption (EXEMPTION_T) – Optional. Exemption status.
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Invalid org unit ID or provided start date later than end date. 403 Forbidden – No permission to see scheduled items for the user in any of the provided org units. 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. When calling this action, you must provide a list of org unit IDs that amount to some or all of the calling user's active enrollments. You can also filter the data retrieved based on its **completion status** or whether an item's due date falls within a provided time window. Exempted items do not have due dates. Additionally, you can filter the data on its **exemption status**. Combining this with `startDateTime` or `endDateTime` will return exempted items within the due date, even though the items are not due.

Note that the calling user must have permission to see user progress for users, manage content, and see the classlist for `_every_` org unit in the provided `orgUnitIdsCSV` list.

Return. This action returns an **ObjectListPage** JSON block containing a list of **ScheduledItem** blocks, sorted by the earliest of start date, end date, or due date that match the provided query parameter filters.

This action ignores invalid selection data on input (unknown or inaccessible org units or items), returning only those items that the user provided has access to.

GET /d2l/api/le/(version)/content/myItems/due/

Retrieve the calling user's scheduled items still due.

Parameters:	version (D2LVERSION) – API version.
Query Parameters:	completion (COMPLETION_T) – Optional. Completion status. orgUnitIdsCSV (CSV of D2LIDs) – List of org unit IDs (limited to 100 or fewer). startDateTime (UTCDateTime) – Optional. Start of time window to examine. endDateTime (UTCDateTime) – Optional. End of time window to examine.
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Invalid org unit ID or provided start date later than end date. 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. When calling this action, you must provide a list of org unit IDs that amount to some or all of the calling user's active enrollments. You can also filter the data retrieved based on its **completion status** or whether an item's due date falls within a provided time window.

Return. This action returns an **ObjectListPage** JSON block containing a list of **ScheduledItem** blocks of the user's scheduled items that are still due, sorted by the due date, and that match the provided query parameter filters. Exempted items are considered not due and will not appear in the result list.

This action ignores invalid selection data on input (unknown or inaccessible org units or items), returning only those items that the calling user has access to.

GET /d2l/api/le/(version)/content/myItems/itemCounts/

Retrieve the quantities of the calling user's scheduled items, organized by org unit.

Parameters:	version (D2LVERSION) – API version.
Query Parameters:	completion (COMPLETION_T) – Optional. Completion status. orgUnitIdsCSV (CSV of D2LIDs) – List of org unit IDs (limited to 100 or fewer). startDateTime (UTCDateTime) – Optional. Start of time window to examine. endDateTime (UTCDateTime) – Optional. End of time window to examine.
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Invalid org unit ID or provided start date later than end date. 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. When calling this action, you must provide a list of org unit IDs that amount to some or all of the calling user's active enrollments. You can also filter the data retrieved based on its **completion status** or whether an item's due date falls within a provided time window. Exempted items do not have due dates.

Return. This action returns an **ObjectListPage** JSON block containing a list of **ScheduledItemCount** blocks containing only the quantity of the calling user's scheduled items that match the provided query parameter filters.

This action ignores invalid selection data on input (unknown or inaccessible org units or items), returning only those items that the calling user has access to.

GET /d2l/api/le/(version)/content/myItems/due/itemCounts/

Retrieve the quantities of the calling user's scheduled items still due.

Parameters:	version (D2LVERSION) – API version.
Query Parameters:	completion (COMPLETION_T) – Optional. Completion status. orgUnitIdsCSV (CSV of D2LIDs) – List of org unit IDs (limited to 100 or fewer). startDateTime (UTCDateTime) – Optional. Start of time window to examine. endDateTime (UTCDateTime) – Optional. End of time window to examine.
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Invalid org unit ID or provided start date later than end date. 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. When calling this action, you must provide a list of org unit IDs that amount to some or all of the calling user's active enrollments. You can also filter the data retrieved based on its **completion status** or whether an item's due date falls within a provided time window.

Return. This action returns an **ObjectListPage** JSON block containing a list of **ScheduledItemCount** blocks containing only the quantity of the calling user's scheduled items that are still due and match the provided query parameter filters. Exempted items are considered not due and will not be counted.

GET /d2l/api/le/(version)/content/myItems/completions/

Retrieve the calling user's completed scheduled items.

Parameters:	version (D2LVERSION) – API version.
Query Parameters:	orgUnitIdsCSV (CSV of D2LIDs) – List of org unit IDs (limited to 100 or fewer). completedFromDateTime (UTCDateTime) – Optional. Start of time window to examine. completedToDateTime (UTCDateTime) – Optional. End of time window to examine.
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Invalid org unit ID or provided start date later than end date. 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. When calling this action, you must provide a list of org unit IDs that amount to some or all of the calling user's active enrollments. You can also filter the data retrieved based on item's completion date, which falls within the provided time window.

Return. This action returns an **ObjectListPage** JSON block containing a list of completed **ScheduledItem** blocks, sorted by completion date that match the provided query parameter filters.

This action ignores invalid selection data on input (unknown or inaccessible org units or items), returning only those items that the calling user has access to.

GET /d2l/api/le/(version)/content/myItems/completions/due/

Retrieve the calling user's completed scheduled items that have a due date.

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

Query Parameters: **orgUnitIdsCSV (CSV of D2LIDs)** – List of org unit IDs (limited to 100 or fewer).
completedFromDateTime (UTCDateTime) – Optional. Start of time window to examine.
completedToDateTime (UTCDateTime) – Optional. End of time window to examine.

Status Codes: **200 OK** – Action succeeded.
400 Bad Request – Invalid org unit ID or provided start date later than end date.
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. When calling this action, you must provide a list of org unit IDs that amount to some or all of the calling user's active enrollments. You can also filter the data retrieved based on item's completion date, which falls within the provided time window.

Return. This action returns an **ObjectListPage** JSON block containing a list of completed **ScheduledItem** blocks, sorted by completion date that match the provided query parameter filters. Exempted items are considered not due and will not appear in the result list.

This action ignores invalid selection data on input (unknown or inaccessible org units or items), returning only those items that the calling user has access to.

GET /d2l/api/le/(version)/(orgUnitId)/content/myItems/

Retrieve the calling user's scheduled items for a particular org unit.

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

Query Parameters: **completion (COMPLETION_T)** – Optional. Completion status.
startDateTime (UTCDateTime) – Optional. Start of time window to examine.
endDateTime (UTCDateTime) – Optional. End of time window to examine.

Status Codes: **200 OK** – Action succeeded.
400 Bad Request – Provided start date later than end date.
403 Forbidden – No permission to see scheduled items for org unit.
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. When calling this action, you can filter the data retrieved based on its **completion status** or whether an item's due date falls within a provided time window. Exempted items do not have due dates.

Return. This action returns an **ObjectListPage** JSON block containing a list of **ScheduledItem** blocks, sorted by the earliest of start date, end date, or due date, that match the provided query parameter filters.

GET /d2l/api/le/(version)/(orgUnitId)/content/myItems/due/

Retrieve the calling user's scheduled items still due for a particular org unit.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID.
Query Parameters:	completion (COMPLETION_T) – Optional. Completion status. startDateTime (UTCDateTime) – Optional. Start of time window to examine. endDateTime (UTCDateTime) – Optional. End of time window to examine.
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Provided start date later than end date. 403 Forbidden – No permission to see scheduled items for org unit. 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. When calling this action, you can filter the data retrieved based on its **completion status** or whether an item's due date falls within a provided time window.

Return. An **ObjectListPage** JSON block containing a list of **ScheduledItem** blocks of the user's scheduled items that are still due, sorted by due date, and that match the provided query parameter filters. Exempted items are considered not due and will not appear in the result list.

GET /d2l/api/le/(version)/(orgUnitId)/content/myItems/itemCount

Retrieve the quantity of the calling user's scheduled items for provided org unit.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID.
Query Parameters:	completion (COMPLETION_T) – Optional. Completion status. startDateTime (UTCDateTime) – Optional. Start of time window to examine. endDateTime (UTCDateTime) – Optional. End of time window to examine.
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Provided start date later than end date. 403 Forbidden – No permission to see scheduled items for org unit. 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. When calling this action, you can filter the data retrieved based on its **completion status** or whether an item's due date falls within a provided time window. Exempted items do not have due dates.

Return. This action returns a **ScheduledItemCount** JSON block containing only the quantity of the calling user's scheduled items that match the provided query parameter filters.

GET /d2l/api/le/(version)/(orgUnitId)/content/myItems/due/itemCount

Retrieve quantity of the calling user's scheduled items still due for a particular org unit.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID.
Query Parameters:	completion (COMPLETION_T) – Optional. Completion status. startDateTime (UTCDateTime) – Optional. Start of time window to examine.

Status Codes:**endTimeTime (UTCDateTime)** – Optional. End of time window to examine.**200 OK** – Action succeeded.**400 Bad Request** – Provided start date later than end date.**403 Forbidden** – No permission to see scheduled items for org unit.**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. When calling this action, you can filter the data retrieved based on its **completion status** or whether an item's due date falls within a provided time window.

Return. This action returns a **ScheduledItemCount** JSON block containing only the quantity of the calling user's scheduled items that are still due and match the provided query parameter filters. Exempted items are considered not due and will not be counted.

GET /d2l/api/le/(version)/overdueItems/

Retrieve the overdue items for a particular user in a particular org unit.

Parameters:**version (D2LVERSION)** – API version.**Query Parameters:****userId (D2LID)** – User for which to retrieve the overdue activities.**orgUnitIdsCSV (CSV of D2LIDs)** – Optional. List of Org unit IDs.**Status Codes:****200 OK** – Action succeeded.**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. For this call, you must provide a *userId* query parameter and may provide an *orgUnitIdsCSV* query parameter:

Users can query about their own overdue items.

Auditors can query about overdue items for users they audit.

Users that have *Manage Content* permission over org units (for example, instructors for course offerings) may be able to query about overdue items for users enrolled in those org units (for example, students), but not for overdue items in org units that the user identified by *userId* has *Manage Content* permission over.

Return. This action returns an **ObjectListPage** JSON block containing a list of **OverdueItem** blocks. Exempted items are considered not due and will not be counted.

GET /d2l/api/le/(version)/overdueItems/myItems

Retrieves the calling user's overdue items, within a number of org units.

Parameters:**version (D2LVERSION)** – API version.**Query Parameters:****orgUnitIdsCSV (CSV of D2LIDs)** – Optional. List of Org unit IDs.**Status Codes:****200 OK** – Action succeeded.**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. You may provide *orgUnitIdsCSV* query parameter for this call; you will be unable to retrieve overdue items for org units for which you do not have *Manage Content* privileges.

Return. This action returns an **ObjectListPage** JSON block containing a list of **OverdueItem**. Exempted items are considered not due and will not be counted.

Table of contents

GET /d2l/api/le/(version)/(orgUnitId)/content/bookmarks

Retrieve a list of topics that have been bookmarked.

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

Oauth2 Scopes: *content:toc:read*

Status Codes: **200 OK** – Action succeeded.
 403 Forbidden – No permission to view course content.
 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 JSON array of **Topic** ToC entries.

GET /d2l/api/le/(version)/(orgUnitId)/content/recent

Retrieve a list of the most recently visited topics.

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

Oauth2 Scopes: *content:toc:read*

Status Codes: **200 OK** – Action succeeded.
 403 Forbidden – No permission to view course content.
 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 JSON array of **Topic** ToC entries.

GET /d2l/api/le/(version)/(orgUnitId)/content/toc

Retrieve the table of course content for an org unit.

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

Query Parameters: **ignoreDateRestrictions** (*boolean*) – (Optional) Include content modules and topics where date restriction would otherwise hide them from view.
 ignoreModuleDateRestrictions (*boolean*) – (Optional) Include content modules where date restriction would otherwise hide them from view.
 userId (D2LID) – (Optional) Fetch the table of course content as it would be viewed by this user.
 moduleId (D2LID) – (Optional) ModuleId used for filtering.
 title (*string*) – (Optional) Topic or module title used for filtering

Oauth2 Scopes: *content:toc:read*

Status Codes: **200 OK** – Action succeeded.
 400 Bad Request – Provided user ID not enrolled in the provided org unit.
 403 Forbidden – No permission to view course content, or no permission to view table of content for provided user ID.
 404 Not Found – No module was found with the provided moduleId.
 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. This action's optional *ignoreDateRestrictions* query parameter supports inclusion of content module and topic entries regardless of date restrictions on viewing content:

If *false* (or you don't provide the query parameter) then this action fetches back the table of contents excluding all items restricted from view by date restriction, unless the calling user has permission to view them anyway (for example, administrator users).

If *true*, then this action fetches back the table of contents including modules and topics that would otherwise be restricted from view by date restriction (start or end dates). No special view permissions are required from the calling user, and therefore, students will be allowed to view such modules (and their sub-modules) and topics even if the restrictions are not met.

Note that the *ignoreDateRestrictions* query parameter is supported as of LE API v1.67 and forward and replaces support for the older *ignoreModuleDateRestrictions* (which the back-end service will ignore if used with version 1.67 of this action and forward). In API versions 1.66 and older, callers can use *ignoreModuleDateRestrictions* to ignore date restrictions only around content modules (descendant content topics would remain excluded, unless the calling user has permission to view them anyway, for example, administrator users).

This action's optional *userId* query parameter lets you retrieve the table of course content as it might be viewed by another user (identified by your provided user ID). When used this way, the calling user context's role affects access to this route:

Auditors can use the route for users they audit.

Users with permission to manage content for the org unit (for example, instructors for course offering org units) can use the route for users enrolled in the org unit.

This action's optional *moduleId* query parameter lets you filter on a specific module. If the module exists, its sub-modules and topics will be returned. If not found, the status code will be 404.

This action's optional *title* query parameter lets you filter on module and topic titles. A "match" occurs if your provided *title* query parameter value occurs as a sub-string in a module or topic title and disregards case.

If module title is matched, itself and all its submodules will be returned.

If only a topic title is matched, only the parent module and the matched topic(s) will be returned.

When both *moduleId* and *title* is provided, the title will be queried in the submodules and topics of the found module.

Return. This action returns a **TableOfContents** JSON block.

User Progress

GET `/d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)/completions/`

Retrieve a list of content topic completions for a topic for users the caller has access to.

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

orgUnitId (D2LID) – Org Unit ID.

topicId (D2LID) – Content Topic ID.

Query Parameters: *userId* (D2LID) – Optional. User ID.

isAscendingSort (*boolean*) – Optional. Whether to sort ascending. True by default.

userBookmark (D2LID) – Optional. Bookmark for last user returned.

pageSize – Optional. Number of entries to return in each data page (default is 20).

OAuth2 Scopes: *content:completions:read*

Status Codes: **200 OK** – Action succeeded.

403 Forbidden – No permission to view content topic completion.

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 an **ObjectListPage** JSON block containing a list of **ContentTopicCompletion** blocks, sorted by *userId* and the *isAscendingSort* parameter.

GET

/d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)/completions/users/(userId)

Retrieve a content topic completion for a topic for a particular user.

Parameters: **version (D2LVERSION)** – API version.
 orgUnitId (D2LID) – Org Unit ID.
 topicId (D2LID) – Content Topic ID.
 userId (D2LID) – User ID.

Oauth2 Scopes: *content:completions:read*

Status Codes: **200 OK** – Action succeeded.
 403 Forbidden – No permission to view content topic completion for this user.
 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 **ContentTopicCompletion** JSON block.

GET /d2l/api/le/(version)/(orgUnitId)/content/completions/mycount/

Retrieves the aggregate count of completed and required content topics in an org unit for the calling user.

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

Query Parameters: **level (CONTENTCOMPLETIONLEVEL_T)** – Completion aggregation level.

Status Codes: **200 OK** – Action succeeded.
 400 Bad Request – Invalid level.
 403 Forbidden – User has no permission to access the org unit.
 404 Not Found – 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.

Return. This action returns an **ObjectListPage** JSON block containing a list of **ContentAggregateCompletion** items.

GET /d2l/api/le/(version)/(orgUnitId)/content/completions/

Retrieves the count of completed and required content topics in an org unit for each of the users in the csv.

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

Query Parameters: **ignoreInvalid (boolean)** – If true, prompt the back-end service to filter out and ignore provided *userIdsCSV* values that are unauthorized or non-existent.
 userIdsCSV (CSV of D2LIDs) – List of user IDs (limited to 100 or fewer).

Status Codes: **200 OK** – Action succeeded.
 400 Bad Request – *userIdsCSV* is invalid, empty, or has more than 100 ids.
 403 Forbidden – At least one user is not enrolled in the org unit, or you don't have access to this user in the classlist.
 404 Not Found – 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.

Return. This action returns a JSON block containing a list of **ContentLearnerProgress** items.

GET /d2l/api/le/(version)/content/completions/(userId)/

Retrieves the count of completed and required content topics for a user in each org unit in the csv.

Parameters:	version (D2LVERSION) – API version. userId (D2LID) – User ID.
Query Parameters:	ignoreInvalid (<i>boolean</i>) – If true, prompt the back-end service to filter out and ignore provided <i>orgUnitIdsCSV</i> values that are unauthorized or non-existent. orgUnitIdsCSV (CSV of D2LIDs) – List of org unit IDs (limited to 100 or fewer).
Status Codes:	200 OK – Action succeeded. 400 Bad Request – User is not in the classlist, or <i>orgUnitIdsCSV</i> is invalid, empty, or has more than 100 ids. 403 Forbidden – You don't have access to this user in the classlist, you are missing the <i>ManageContent</i> permission, or the user is not enrolled in one of the org units. 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 JSON block containing a list of **ContentLearnerProgress** items.

GET /d2l/api/le/(version)/(orgUnitId)/content/userprogress/

Retrieve the user progress items in an org unit, for specific users or content topics.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org Unit ID.
Query Parameters:	userId (D2LID) – Report on progress for these users (you can use this parameter more than once for multiple users). objectId (D2LID) – Report on progress through these content topics (you can use this parameter more than once for multiple topics). pageSize – Optional. Number of entries to return in each data page (default is 20).
Status Codes:	200 OK – Action succeeded. 429 Too Many Requests – API call-rate limit exceeded.
API Versions:	unstable – Route first appears in LMS v10.6.0.

Input. The calling user context's role affects the scope of this action:

Users can query about their own progress through content.

Auditors can query about progress for users they audit.

Users with permission to see progress for all users over org units (for example, instructors for course offering org units) may be able to query about progress for all users enrolled in those org units.

Return. This action returns an **ObjectListPage** JSON block containing a list of **UserProgressData**.

Note that with this action, you will only retrieve user progress information for those users that have made some progress through content topics; you will not retrieve records for users that have made no progress on topics past the default state.

GET /d2l/api/le/(version)/(orgUnitId)/content/userprogress/(topicId)

Retrieve one user's progress within an org unit for a particular content topic.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org Unit ID. topicId (D2LID) – Topic ID.
Query Parameters:	userId (D2LID) – Report on progress for this user.
Status Codes:	200 OK – Action succeeded. 429 Too Many Requests – API call-rate limit exceeded.
API Versions:	unstable – Route first appears in LMS v10.6.0.

Input. The calling user context's role affects the scope of this action:

Users can query about their own progress through content.

Auditors can query about progress for a user they audit.

Users with permission to see progress for all users over org units (for example, instructors for course offering org units) may be able to query about progress for all users enrolled in those org units.

Return. This action returns a **UserProgressData** JSON block.

Note that with this action, you will retrieve all user progress records for the topic, including records with a default state for those users that have made no progress past that state.

POST /d2l/api/le/(version)/(orgUnitId)/content/userprogress/

Update a user progress item.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org Unit ID.
JSON Parameters:	UserProgressData (UserProgressData) – Updated progress for a content topic.
Status Codes:	200 OK – Action succeeded. 429 Too Many Requests – API call-rate limit exceeded.
API Versions:	unstable – Route first appears in LMS v10.6.0.

Input. Note that the *UserId* property in your provided user progress data must be the ID for the calling user context: this action only lets users update their *own progress* through content.

Return. This action returns a **UserProgressData** JSON data block containing the updated state of the user's progress.

PUT

/d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)/completions/users/(userId)

Update a content topic completion for a topic for a particular user.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org Unit ID. topicId (D2LID) – Content Topic ID. userId (D2LID) – User ID.
JSON Parameters:	ContentTopicCompletionUpdate (ContentTopicCompletionUpdate) – Updated completion information for the topic.
Oauth2 Scopes:	<i>content:completions:write</i>
Status Codes:	200 OK – Action succeeded. 403 Forbidden – No permission to manage content topic completion for this user. 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. This action expects a **ContentTopicCompletionUpdate** JSON block as input.

Pacing

GET /d2l/api/le/(version)/(orgUnitId)/content/pacing

Retrieve pacing dates for content.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID.
Status Codes:	200 OK – Action succeeded. 403 Forbidden – No permission to view course content. 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 **PacingInfo** JSON data block containing the course and semester start and end dates.

PUT /d2l/api/le/(version)/(orgUnitId)/content/pacing

Set pacing dates for content.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID.
JSON Parameters:	PacingData (PacingData)
Status Codes:	200 OK – Action succeeded. 403 Forbidden – No permission to view course content. 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.

PUT /d2l/api/le/(version)/(orgUnitId)/content/pacing/modules/(moduleId)

Set pacing dates for a specific module.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID. moduleId (D2LID) – Module ID.
JSON Parameters:	ModulePacingData (ModulePacingData)
Status Codes:	200 OK – Action succeeded. 403 Forbidden – No permission to view course content. 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.

Captions

DELETE

/d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)/captions/(language)

Delete the captions file associated with a content topic for a specific language.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID. topicId (D2LID) – Content Topic ID. language (string) – ISO 639-1 language and culture code of the captions file to delete.
Oauth2 Scopes:	<i>content:file:write</i>
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Content object with provided topic ID is not an audio topic or video topic, or invalid language provided. 403 Forbidden – No permission to manage content files. 404 Not Found – No such topic or org unit found, or could not find captions file associated with the provided topic and language. 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. The format of the *language* query parameter must be *(language)-(culture)*, or *(language)* if culture does not apply. This parameter must match the ISO 639-1 code of the desired language exactly. For example, “FR” and “FR-

CA" will be treated as separate resources, and deleting "FR" will not delete "FR-CA".

GET /d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)/captions

Retrieve metadata for all captions associated with an audio or video content topic.

Parameters:

- version (D2LVERSION)** – API version.
- orgUnitId (D2LID)** – Org unit ID.
- topicId (D2LID)** – Content Topic ID.

Oauth2 Scopes: *content:file:write*

Status Codes:

- 200 OK** – Action succeeded.
- 400 Bad Request** – Content object with provided topic ID is not an audio topic or video topic.
- 403 Forbidden** – No permission to manage content files.
- 404 Not Found** – No such topic found in org unit, or 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.

Return. This action returns an **ObjectListPage** JSON block containing a list of **CaptionsMetadata** blocks, sorted by their **LanguageCode** and **LanguageCulture** properties in ascending order.

PUT

/d2l/api/le/(version)/(orgUnitId)/content/topics/(topicId)/captions/(language)

Provide a language-specific captions file associated with a particular content topic.

Parameters:

- version (D2LVERSION)** – API version.
- orgUnitId (D2LID)** – Org unit ID.
- topicId (D2LID)** – Content Topic ID.
- language (string)** – ISO 639-1 language and culture code that the captions file will be associated with.

Oauth2 Scopes: *content:file:write*

Status Codes:

- 200 OK** – Action succeeded.
- 400 Bad Request** – Content object with provided topic ID is not an audio topic or video topic, or invalid language provided, or invalid file type provided.
- 403 Forbidden** – No permission to manage content files.
- 404 Not Found** – No such topic found in org unit, or 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.

Input. Provide an uploaded SRT or WebVTT content file using the **simple file upload** process; the content-disposition part header for the file part should have the name *file*. This file will be saved in a content folder with a name matching the topic file's name, without file extension.

The format of the *language* query parameter must be *(language)-(culture)*, or *(language)* if culture does not apply. This parameter must match the ISO 639-1 code of the desired language exactly. For example, "FR" and "FR-CA" will be treated as separate resources, and uploading to "FR" will not modify "FR-CA".

Note that if no such captions file existed previously for this language, for this content topic, this action will create one; if one already existed, this action will replace that previous file.

Return. This action returns a **CaptionsMetadata** JSON block describing the updated captions file.

POST `/d2l/api/le/(version)/(orgUnitId)/content/order/objectId/(objectId)`

Specify the position order of a content object with respect to its sibling objects.

Parameters:	version (D2LVERSION) – API version. orgUnitId (D2LID) – Org unit ID. objectId (D2LID) – Content Object ID.
Query Parameters:	position (<i>string</i>) – The order position within siblings and can be either <i>first</i> , <i>last</i> , or the <i>objectId</i> to be placed after.
Oauth2 Scopes:	<i>content:modules:manage content:topics:manage</i>
Status Codes:	200 OK – Action succeeded. 400 Bad Request – Content object with provided is invalid or not found. 403 Forbidden – No permission to manage content files. 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 a *position* query parameter value to indicate where in the content sorting you want a target content object (topic or module) to have its position. You may specify one of three kinds of values: *first* to move the specified target content object to be first in order amongst its siblings; *last* to move the target object to be last in order; or, provide a content object ID value to indicate the object you want to be immediately before your target object in the order.

Note that you can only use this action to arrange the order of content objects in a single generation (that is, sibling content objects all with the same parent content module, or all at the root level); you cannot use this action to move an object across a generation boundary (that is, move it from within one content module to within another content module).

« Grades (grade objects, categories, values, schemes) · [home · reference · community | search · index · routing table
 · scopes table] · Course updates for end users »