

# Echo360 Public API Swagger Docs

## Base URLs:

- <https://echo360.org> (US Region)
- <https://echo360.org.uk> (UK Region)
- <https://echo360.org.au> (AU/ANZ Region)
- <https://echo360.ca> (Canada Region)

Version: 1.0

22 August 2017

## Schemes:

# Summary

## Tag: campuses

Operation	Description
<a href="#">GET /public/api/v1/campuses</a>	List of all Campuses for current Institution
<a href="#">POST /public/api/v1/campuses</a>	Add Campus
<a href="#">GET /public/api/v1/campuses/{campusId}</a>	Get Campus by ID or External ID
<a href="#">PUT /public/api/v1/campuses/{campusId}</a>	Update Campus by ID or External ID
<a href="#">DELETE /public/api/v1/campuses/{campusId}</a>	Delete Campus by ID or External ID

## Tag: sections

Operation	Description
<a href="#">GET /public/api/v1/sections</a>	List of all Sections for current Institution
<a href="#">POST /public/api/v1/sections</a>	Add Section
<a href="#">GET /public/api/v1/sections/{sectionId}</a>	Get Section by ID or External ID
<a href="#">PUT /public/api/v1/sections/{sectionId}</a>	Update Section by ID or External ID
<a href="#">DELETE /public/api/v1/sections/{sectionId}</a>	Delete Section by ID or External ID
<a href="#">POST /public/api/v1/sections/{sectionId}/clone</a>	Clone Section by ID or External ID
<a href="#">PUT /public/api/v1/sections/{sectionId}/lms-course-ids</a>	Update Section LMS Course IDs by ID or External ID
<a href="#">PUT /public/api/v1/sections/{sectionId}/secondary-instructors</a>	Update Section Secondary Instructors by ID or External ID

## Tag: rooms

Operation	Description
GET /public/api/v1/rooms	List of all Rooms for current Institution
POST /public/api/v1/rooms	Add Room
GET /public/api/v1/rooms/{roomId}	Get Room by ID or External ID
PUT /public/api/v1/rooms/{roomId}	Update Room by ID or External ID
DELETE /public/api/v1/rooms/{roomId}	Delete Room by ID or External ID

## Tag: lms-profiles

Operation	Description
GET /public/api/v1/lms-profiles	List of all LmsProfiles for current Institution
POST /public/api/v1/lms-profiles	Add LmsProfile by externalSystemIdRef
GET /public/api/v1/lms-profiles/{lmsProfileId}	Get LmsProfile by Id
PUT /public/api/v1/lms-profiles/{lmsProfileId}	Update LmsProfile
DELETE /public/api/v1/lms-profiles/{lmsProfileId}	Delete a LmsProfile

## Tag: organizations

Operation	Description
GET /public/api/v1/organizations	List of all Organizations for current Institution
POST /public/api/v1/organizations	Add Organization
GET /public/api/v1/organizations/{organizationId}	Get Organization by ID or External ID
PUT /public/api/v1/organizations/{organizationId}	Update Organization by ID or External ID
DELETE /public/api/v1/organizations/{organizationId}	Delete Organization by ID or External ID

## Tag: users

Operation	Description
GET /public/api/v1/users	List of all Users for current Institution
POST /public/api/v1/users	Add User
GET /public/api/v1/users/{userId}	Get User by ID, Email, or External ID
PUT /public/api/v1/users/{userId}	Update User by ID, Email, or External ID

GET /public/api/v1/users/{userId}/invite	Resend User Invite by ID, Email, or External ID
GET /public/api/v1/users/{userId}/status/{status}	Disable / Enable User by ID, Email, or External ID

## Tag: lessons

Operation	Description
GET /public/api/v1/lessons/{lessonId}	Get existing Lesson by ID
PUT /public/api/v1/lessons/{lessonId}	Update existing Lesson by ID
DELETE /public/api/v1/lessons/{lessonId}	Delete existing Lesson by ID
GET /public/api/v1/sections/{sectionId}/lessons	List of all existing and future Lessons for current Section
POST /public/api/v1/sections/{sectionId}/lessons	Add Lesson

## Tag: terms

Operation	Description
GET /public/api/v1/terms	List of all Terms for current Institution
POST /public/api/v1/terms	Add Term
GET /public/api/v1/terms/{termId}	Get Term by ID or External ID
PUT /public/api/v1/terms/{termId}	Update Term by ID or External ID
DELETE /public/api/v1/terms/{termId}	Delete Term by ID or External ID

## Tag: buildings

Operation	Description
GET /public/api/v1/buildings	List of all Buildings for current Institution
POST /public/api/v1/buildings	Add Building
GET /public/api/v1/buildings/{buildingId}	Get Building by ID or External ID
PUT /public/api/v1/buildings/{buildingId}	Update Building by ID or External ID
DELETE /public/api/v1/buildings/{buildingId}	Delete Building by ID or External ID

## Tag: publish

Operation	Description
GET /public/api/v1/publish/media/lessons/{lessonId}	list publishable media for a lesson

GET /public/api/v1/publish/media/{mediaId}/lessons	list all lessons a media is published to
POST /public/api/v1/publish/media/{mediaId}/lessons/{lessonId}	publish media to a lesson and with availability set by json
PUT /public/api/v1/publish/media/{mediaId}/lessons/{lessonId}	update the status of a media that's been published to a lesson
DELETE /public/api/v1/publish/media/{mediaId}/lessons/{lessonId}	unpublish media from a section

## Tag: schedules

Operation	Description
GET /public/api/v1/schedules	List of all Schedules for current Institution
POST /public/api/v1/schedules	Add Schedule
GET /public/api/v1/schedules/{scheduleId}	Get Schedule by ID or External ID
PUT /public/api/v1/schedules/{scheduleId}	Update Schedule by ID or External ID
DELETE /public/api/v1/schedules/{scheduleId}	Delete Schedule by ID or External ID
PUT /public/api/v1/schedules/{scheduleId}/deviceOptions	Update Schedule's Device Options by ID or External ID
PUT /public/api/v1/schedules/{scheduleId}/external_id	Update Schedule's External ID by ID or External ID
PUT /public/api/v1/schedules/{scheduleId}/flags	Update Schedule's Flags by ID or External ID
PUT /public/api/v1/schedules/{scheduleId}/instructors	Update Schedule's Instructors by ID or External ID
PUT /public/api/v1/schedules/{scheduleId}/name	Update Schedule's Name by ID or External ID
PUT /public/api/v1/schedules/{scheduleId}/section	Update Schedule's Section by ID or External ID
PUT /public/api/v1/schedules/{scheduleId}/times	Update Schedule's Times by ID or External ID

## Tag: courses

Operation	Description
GET /public/api/v1/courses	List of all Courses for current Institution
POST /public/api/v1/courses	Add Course
GET /public/api/v1/courses/{courseId}	Get Course by ID or External ID
PUT /public/api/v1/courses/{courseId}	Update Course By ID or External ID
DELETE /public/api/v1/courses/{courseId}	Delete Course By ID or External ID

## Tag: departments

Operation	Description
GET /public/api/v1/departments	List of all Departments for current Institution
POST /public/api/v1/departments	Add Department
GET /public/api/v1/departments/{departmentId}	Get Department by ID or External ID
PUT /public/api/v1/departments/{departmentId}	Update Department by ID or External ID
DELETE /public/api/v1/departments/{departmentId}	Delete Department by ID or External ID

## Tag: enrollment

Operation	Description
GET /public/api/v1/sections/{sectionId}/users	List of all User enrollments for Section
POST /public/api/v1/sections/{sectionId}/users	Enroll User in a Section
GET /public/api/v1/sections/{sectionId}/users/{userId}	Get User enrollment for Section
PUT /public/api/v1/sections/{sectionId}/users/{userId}	Update a User's Section Role
DELETE /public/api/v1/sections/{sectionId}/users/{userId}	Unenroll User from Section

## Tag: oauth2access\_token

Operation	Description
POST /oauth2/access_token	Request an access token

## Tag: room-configuration

Operation	Description
GET /public/api/v1/rooms/{roomId}/device-access	Get Network Connection Protocol and Port for a given Room
GET /public/api/v1/rooms/{roomId}/hardware1/inputs	Get Hardware 1 (SCHD) Input Configuration
GET /public/api/v1/rooms/{roomId}/hardware2/front-panel	Get the enabled status of a Hardware 2 (Echo360 PRO) front panel
GET /public/api/v1/rooms/{roomId}/hardware2/inputs	Get Hardware 2 (Echo360 PRO) Input Configuration
GET	Get the One Touch details of a front-panel-enabled

<a href="#">/public/api/v1/rooms/{roomId}/hardware2/one-touch</a>	Hardware 2 (Echo360 PRO) device
GET <a href="#">/public/api/v1/rooms/{roomId}/hardware3/front-panel</a>	Get the enabled status of a Hardware 3 (Echo360 POD) front panel
GET <a href="#">/public/api/v1/rooms/{roomId}/hardware3/inputs</a>	Get Hardware 3 (Echo360 POD) Input Configuration
GET <a href="#">/public/api/v1/rooms/{roomId}/hardware3/one-touch</a>	Get the One Touch details of a front-panel-enabled Hardware 3 (Echo360 POD) device
GET <a href="#">/public/api/v1/rooms/{roomId}/local-user/admin</a>	Get Local Admin User Credentials
GET <a href="#">/public/api/v1/rooms/{roomId}/local-user/generic</a>	Get Local Generic User Credentials
GET <a href="#">/public/api/v1/rooms/{roomId}/network</a>	Get Network Settings plus Time Servers for a given Room
GET <a href="#">/public/api/v1/rooms/{roomId}/outbound-proxy</a>	Get Outbound Proxy details for a given Room
GET <a href="#">/public/api/v1/rooms/{roomId}/software1/inputs</a>	Get Software (CCAP) Input Configuration

## Tag: lti-links

Operation	Description
POST <a href="#">/public/api/v1/lms/{lmsProfileId}/contexts/{contextId}/links</a>	Create LTI Link
GET <a href="#">/public/api/v1/lms/{lmsProfileId}/contexts/{contextId}/links/{linkId}</a>	Get LTI Link by ID
PUT <a href="#">/public/api/v1/lms/{lmsProfileId}/contexts/{contextId}/links/{linkId}</a>	Update LTI Link
DELETE <a href="#">/public/api/v1/lms/{lmsProfileId}/contexts/{contextId}/links/{linkId}</a>	Delete LTI Link by ID
GET <a href="#">/public/api/v1/sections/{sectionId}/lti-links</a>	List all LTI Links for a single section

## Paths

### POST [/oauth2/access\\_token](#)

Tags: [oauth2access\\_token](#)

Request an access token

#### DESCRIPTION

## REQUEST BODY

application/x-www-form-urlencoded

## REQUEST PARAMETERS

Name	Description	Type	Data type	
grant_type	Can be one of the following: client_credentials, password, or refresh_token	query	string	required
client_id	Needed for all grant types	query	string	required
client_secret	Needed for all grant types except refresh_token	query	string	
username	If password chosen	query	string	
password	If password chosen	query	string	
refresh_token	If refresh_token chosen	query	string	

## RESPONSES

### 200 OK

Successful

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/buildings

Tags: [buildings](#)

List of all Buildings for current Institution

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type
campusId	ID or External ID of the Campus to filter Buildings	query	<i>string</i> (default: "None")
limit	Number of results	query	<i>integer</i> (int32)
offset	Key returned from prior result to get next page	query	<i>string</i>

## RESPONSES

### 200 OK

Successful

#### ITEMS

*Building*

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## POST /public/api/v1/buildings

Add Building

Tags: [buildings](#)

### DESCRIPTION

### REQUEST BODY

Building data

CreateUpdateBuilding

## RESPONSES

### 200 OK

Successful

Building

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 500 Internal Server Error

Internal Error

## DELETE /public/api/v1/buildings/{buildingId}

Tags: [buildings](#)

Delete Building by ID or External ID

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
buildingId	ID or External ID of the Building to Delete	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

### 400 Bad Request

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**500 Internal Server Error**

Internal Error

**GET /public/api/v1/buildings/{buildingId}**

Get Building by ID or External ID

Tags: [buildings](#)

**DESCRIPTION**

**REQUEST PARAMETERS**

Name	Description	Type	Data type	
buildingId	ID or External ID of the Building to Get	path	<i>string</i>	<b>required</b>

**RESPONSES**

**200 OK**

Successful

[Building](#)

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**500 Internal Server Error**

Internal Error

## PUT /public/api/v1/buildings/{buildingId}

Update Building by ID or External ID

Tags: [buildings](#)

### DESCRIPTION

### REQUEST BODY

Building data

[CreateUpdateBuilding](#)

### REQUEST PARAMETERS

Name	Description	Type	Data type	
buildingId	ID or External ID of the Building to Update	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[Building](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## GET /public/api/v1/campuses

List of all Campuses for current Institution

Tags: [campuses](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type
limit	Number of results	query	<i>integer (int32)</i>
offset	Key returned from prior result to get next page	query	<i>string</i>

### RESPONSES

#### 200 OK

Successful

#### ITEMS

[Campus](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## POST /public/api/v1/campuses

Add Campus

Tags: [campuses](#)

## DESCRIPTION

## REQUEST BODY

Campus data

```
CreateUpdateCampus
```

## RESPONSES

### 200 OK

Successful

```
Campus
```

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 500 Internal Server Error

Internal Error

## DELETE /public/api/v1/campuses/{campusId}

Delete Campus by ID or External ID

Tags: [campuses](#)

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type
campusId	ID or External ID of the Campus to Delete	path	string <span>required</span>

## RESPONSES

### 200 OK

Successful

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/campuses/{campusId}

Get Campus by ID or External ID

Tags: [campuses](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
campusId	ID or External ID of the Campus to Get	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

Campus

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## PUT /public/api/v1/campuses/{campusId}

Update Campus by ID or External ID

Tags: [campuses](#)

### DESCRIPTION

### REQUEST BODY

Campus data

```
CreateUpdateCampus
```

### REQUEST PARAMETERS

Name	Description	Type	Data type	
campusId	ID or External ID of the Campus to Update	path	string	<b>required</b>

### RESPONSES

#### 200 OK

Successful

```
Campus
```

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/courses

List of all Courses for current Institution

Tags: [courses](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type
limit	Number of results	query	<i>integer (int32)</i>
offset	Key returned from prior result to get next page	query	<i>string</i>

### RESPONSES

#### 200 OK

Successful

#### ITEMS

[Course](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

**405 Method Not Allowed**

Method Not Allowed

**500 Internal Server Error**

Internal Error

**POST /public/api/v1/courses**

Add Course

Tags: [courses](#)

**DESCRIPTION**

**REQUEST BODY**

Course data

[CreateUpdateCourse](#)

**RESPONSES**

**200 OK**

Successful

[Course](#)

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**500 Internal Server Error**

Internal Error

## DELETE /public/api/v1/courses/{courseId}

Delete Course By ID or External ID

Tags: [courses](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
courseId	ID or External ID of the Course to Delete	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 500 Internal Server Error

Internal Error

## GET /public/api/v1/courses/{courseId}

Get Course by ID or External ID

Tags: [courses](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
courseId	ID or External ID of the Course to Get	path	<i>string</i>	<b>required</b>

### RESPONSES

## 200 OK

Successful

Course

## 400 Bad Request

Bad Request

## 401 Unauthorized

Not Authenticated

## 403 Forbidden

Forbidden

## 404 Not Found

Resource Not Found

## 405 Method Not Allowed

Method Not Allowed

## 500 Internal Server Error

Internal Error

## PUT /public/api/v1/courses/{courseId}

Update Course By ID or External ID

Tags: [courses](#)

### DESCRIPTION

### REQUEST BODY

Course Data

CreateUpdateCourse

### REQUEST PARAMETERS

Name	Description	Type	Data type	
courseId	ID or External ID of the Course to Update	path	string	required

### RESPONSES

**200 OK**

Successful

[Course](#)**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**405 Method Not Allowed**

Method Not Allowed

**500 Internal Server Error**

Internal Error

**GET /public/api/v1/departments**

List of all Departments for current Institution

Tags: [departments](#)**DESCRIPTION****REQUEST PARAMETERS**

Name	Description	Type	Data type
organizationId	ID or External ID of the Organization to filter Departments	query	<i>string</i> (default: "None")
limit	Number of results	query	<i>integer</i> (int32)
offset	Key returned from prior result to get next page	query	<i>string</i>

**RESPONSES****200 OK**

Successful

## ITEMS

*Department*

### **400 Bad Request**

Bad Request

### **401 Unauthorized**

Not Authenticated

### **403 Forbidden**

Forbidden

### **404 Not Found**

Resource Not Found

### **405 Method Not Allowed**

Method Not Allowed

### **500 Internal Server Error**

Internal Error

## **POST /public/api/v1/departments**

Add Department

Tags: [departments](#)

### **DESCRIPTION**

### **REQUEST BODY**

Department data

[CreateUpdateDepartment](#)

### **RESPONSES**

#### **200 OK**

Successful

[Department](#)

#### **400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**500 Internal Server Error**

Internal Error

**DELETE /public/api/v1/departments/{departmentId}**

Delete Department by ID or External ID

Tags: [departments](#)**DESCRIPTION****REQUEST PARAMETERS**

Name	Description	Type	Data type	
departmentId	ID or External ID of the Department to Delete	path	string	<b>required</b>

**RESPONSES****200 OK**

Successful

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**500 Internal Server Error**

Internal Error

## GET /public/api/v1/departments/{departmentId}

Get Department by ID or External ID

Tags: [departments](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
departmentId	ID of the Department to Get	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[Department](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## PUT /public/api/v1/departments/{departmentId}

Update Department by ID or External ID

Tags: [departments](#)

### DESCRIPTION

### REQUEST BODY

Department data

CreateUpdateDepartment

#### REQUEST PARAMETERS

Name	Description	Type	Data type
departmentId	ID or External ID of the Department to Update	path	string <span>required</span>

#### RESPONSES

##### 200 OK

Successful

Department

##### 400 Bad Request

Bad Request

##### 401 Unauthorized

Not Authenticated

##### 403 Forbidden

Forbidden

##### 404 Not Found

Resource Not Found

##### 405 Method Not Allowed

Method Not Allowed

##### 500 Internal Server Error

Internal Error

#### DELETE /public/api/v1/lessons/{lessonId}

Delete existing Lesson by ID

Tags: [lessons](#)

#### DESCRIPTION

#### REQUEST PARAMETERS

Name	Description	Type	Data type
------	-------------	------	-----------

lessonId ID of the existing Lesson to Delete path *string* **required**

## RESPONSES

### 200 OK

Successful

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/lessons/{lessonId}

Get existing Lesson by ID

Tags: [lessons](#)

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
lessonId	ID of the existing Lesson to Get	path	<i>string</i>	<b>required</b>

## RESPONSES

### 200 OK

Successful

[Lesson](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## PUT /public/api/v1/lessons/{lessonId}

Update existing Lesson by ID

Tags: [lessons](#)

### DESCRIPTION

### REQUEST BODY

Lesson data

[LessonInfo](#)

### REQUEST PARAMETERS

Name	Description	Type	Data type	
lessonId	ID of the existing Lesson to Update	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[Lesson](#)

#### 400 Bad Request

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**405 Method Not Allowed**

Method Not Allowed

**500 Internal Server Error**

Internal Error

**GET /public/api/v1/lms-profiles**

List of all LmsProfiles for current Institution

Tags: [lms-profiles](#)**DESCRIPTION****REQUEST PARAMETERS**

Name	Description	Type	Data type
limit	Number of results	query	<i>integer</i> (int32)

**RESPONSES****200 OK**

Successful

**ITEMS**[LmsProfile](#)**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**405 Method Not Allowed**

Method Not Allowed

**500 Internal Server Error**

Internal Error

**POST /public/api/v1/lms-profiles**

Add LmsProfile by externalSystemIdRef

Tags: [lms-profiles](#)

**DESCRIPTION**

**REQUEST BODY**

LmsProfile data

[LmsProfileInfo](#)

**RESPONSES**

**200 OK**

Successful

[LmsProfile](#)

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**405 Method Not Allowed**

Method Not Allowed

**500 Internal Server Error**

Internal Error

## DELETE /public/api/v1/lms-profiles/{lmsProfileId}

Delete a LmsProfile

Tags: [lms-profiles](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
lmsProfileId	Name of the LmsProfile to Deleted	path	string	<b>required</b>

### RESPONSES

#### 200 OK

Successful

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 500 Internal Server Error

Internal Error

## GET /public/api/v1/lms-profiles/{lmsProfileId}

Get LmsProfile by Id

Tags: [lms-profiles](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
lmsProfileId	ID of the LmsProfile to Get	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

[LmsProfile](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 500 Internal Server Error

Internal Error

## PUT /public/api/v1/lms-profiles/{lmsProfileId}

Update LmsProfile

Tags: [lms-profiles](#)

### DESCRIPTION

### REQUEST BODY

LmsProfile data

[LmsProfileInfo](#)

### REQUEST PARAMETERS

Name	Description	Type	Data type	
lmsProfileId	ID of the LmsProfile to Update	path	<i>string</i>	<b>required</b>

## RESPONSES

### 200 OK

Successful

[LmsProfile](#)

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**405 Method Not Allowed**

Method Not Allowed

**500 Internal Server Error**

Internal Error

**POST /public/api/v1/lms/{lmsProfileId}/contexts/{contextId}/links**

Create LTI Link

Tags: [lti-links](#)

**DESCRIPTION**

**REQUEST BODY**

LTI Link Info

[LtiLinkInfo](#)

**REQUEST PARAMETERS**

Name	Description	Type	Data type	
lmsProfileId	ID of the LMS Profile	path	string	<b>required</b>
contextId	ID of the LTI Context	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

[LtiLink](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 500 Internal Server Error

Internal Error

## DELETE

**/public/api/v1/lms/{lmsProfileId}/contexts/{contextId}/links/{linkId}**

Delete LTI Link by ID

Tags: [lti-links](#)

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
lmsProfileId	ID of the LMS Profile	path	<i>string</i>	<b>required</b>
contextId	ID of the LTI Context	path	<i>string</i>	<b>required</b>
linkId	ID of the LTI Resource Link	path	<i>string</i>	<b>required</b>

## RESPONSES

### 200 OK

Successful

[LtiLink](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 500 Internal Server Error

Internal Error

### GET /public/api/v1/lms/{lmsProfileId}/contexts/{contextId}/links/{linkId} Get LTI Link by ID

Tags: [lti-links](#)

#### DESCRIPTION

#### REQUEST PARAMETERS

Name	Description	Type	Data type	
lmsProfileId	ID of the LMS Profile	path	string	required
contextId	ID of the LTI Context	path	string	required
linkId	ID of the LTI Resource Link	path	string	required

#### RESPONSES

##### 200 OK

Successful

[LtiLink](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 500 Internal Server Error

Internal Error

## PUT /public/api/v1/lms/{lmsProfileId}/contexts/{contextId}/links/{linkId}

Update LTI Link

Tags: [lti-links](#)

### DESCRIPTION

### REQUEST BODY

LTI Link Info

[LtiLinkInfo](#)

### REQUEST PARAMETERS

Name	Description	Type	Data type	
lmsProfileId	ID of the LMS Profile	path	string	<b>required</b>
contextId	ID of the LTI Context	path	string	<b>required</b>
linkId	ID of the LTI Resource Link	path	string	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[LtiLink](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 500 Internal Server Error

Internal Error

## GET /public/api/v1/organizations

Tags: [organizations](#)

List of all Organizations for current Institution

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type
limit	Number of results	query	<i>integer (int32)</i>
offset	Key returned from prior result to get next page	query	<i>string</i>

### RESPONSES

#### 200 OK

Successful

#### ITEMS

[Organization](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## POST /public/api/v1/organizations

Add Organization

Tags: [organizations](#)

### DESCRIPTION

### REQUEST BODY

Organization data

```
CreateUpdateOrganization
```

### RESPONSES

#### 200 OK

Successful

```
Organization
```

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 500 Internal Server Error

Internal Error

## DELETE /public/api/v1/organizations/{organizationId} Delete Organization by ID or External ID

Tags: [organizations](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type
------	-------------	------	-----------

organizationId ID or External ID of the Organization to Delete path *string* **required**

## RESPONSES

### 200 OK

Successful

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/organizations/{organizationId}

Get Organization by ID or External ID

Tags: [organizations](#)

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type
organizationId	ID or External ID of the Organization to Get	path	<i>string</i> <b>required</b>

## RESPONSES

### 200 OK

Successful

[Organization](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**405 Method Not Allowed**

Method Not Allowed

**500 Internal Server Error**

Internal Error

**PUT /public/api/v1/organizations/{organizationId}**

Update Organization by ID or External ID

Tags: [organizations](#)

**DESCRIPTION**

**REQUEST BODY**

Organization data

[CreateUpdateOrganization](#)

**REQUEST PARAMETERS**

Name	Description	Type	Data type	
organizationId	ID or External ID of the Organization to Update	path	string	<b>required</b>

**RESPONSES**

**200 OK**

Successful

[Organization](#)

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**405 Method Not Allowed**

Method Not Allowed

**500 Internal Server Error**

Internal Error

**GET /public/api/v1/publish/media/lessons/{lessonId}**

list publishable media for a lesson

Tags: [publish](#)

**DESCRIPTION**

**REQUEST PARAMETERS**

Name	Description	Type	Data type	
lessonId	ID of the lesson	path	string	<b>required</b>

**RESPONSES**

**200 OK**

Successful

**400 Bad Request**

Not Authenticated

**500 Internal Server Error**

Internal Error

**GET /public/api/v1/publish/media/{mediaId}/lessons**

list all lessons a media is published to

Tags: [publish](#)

**DESCRIPTION**

## REQUEST PARAMETERS

Name	Description	Type	Data type	
mediaId	ID of the media	path	string	required

## RESPONSES

### 200 OK

Successful

### 400 Bad Request

Not Authenticated

### 500 Internal Server Error

Internal Error

## DELETE

`/public/api/v1/publish/media/{mediaId}/lessons/{lessonId}`

unpublish media from a section

Tags: [publish](#)

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
mediaId	ID of the media	path	string	required
lessonId	ID of the lesson	path	string	required

## RESPONSES

### 200 OK

Successful

### 400 Bad Request

Not Authenticated

### 500 Internal Server Error

Internal Error

## POST

publish media to a lesson and with availability set by json

**/public/api/v1/publish/media/{mediaId}/lessons/{lessonId}**

Tags: [publish](#)

### DESCRIPTION

### REQUEST BODY

Availability

[Availability](#)

### REQUEST PARAMETERS

Name	Description	Type	Data type	
mediaId	ID of the media	path	<i>string</i>	<b>required</b>
lessonId	ID of the lesson	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

#### 400 Bad Request

Not Authenticated

#### 500 Internal Server Error

Internal Error

## PUT

update the status of a media that's been published to a lesson

**/public/api/v1/publish/media/{mediaId}/lessons/{lessonId}**

Tags: [publish](#)

### DESCRIPTION

### REQUEST BODY

Availability

## Availability

### REQUEST PARAMETERS

Name	Description	Type	Data type	
mediaId	ID of the media	path	<i>string</i>	<b>required</b>
lessonId	ID of the lesson	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

#### 400 Bad Request

Not Authenticated

#### 500 Internal Server Error

Internal Error

## GET /public/api/v1/rooms

List of all Rooms for current Institution

Tags: [rooms](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type
buildingId	ID or External ID of the Building to filter Rooms	query	<i>string</i> (default: "None")
limit	Number of results	query	<i>integer</i> (int32)
offset	Key returned from prior result to get next page	query	<i>string</i>

### RESPONSES

**200 OK**  
Successful

**ITEMS**

[Room](#)

**400 Bad Request**  
Bad Request

**401 Unauthorized**  
Not Authenticated

**403 Forbidden**  
Forbidden

**404 Not Found**  
Resource Not Found

**405 Method Not Allowed**  
Method Not Allowed

**500 Internal Server Error**  
Internal Error

**POST /public/api/v1/rooms**

Add Room

Tags: [rooms](#)

**DESCRIPTION**

**REQUEST BODY**

Room data

[CreateUpdateRoom](#)

**RESPONSES**

**200 OK**  
Successful

[Room](#)

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**500 Internal Server Error**

Internal Error

**DELETE /public/api/v1/rooms/{roomId}**

Delete Room by ID or External ID

Tags: [rooms](#)

**DESCRIPTION**

**REQUEST PARAMETERS**

Name	Description	Type	Data type
roomId	ID or External ID of the Room to Delete	path	<i>string</i> <b>required</b>

**RESPONSES**

**200 OK**

Successful

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**500 Internal Server Error**

Internal Error

## GET /public/api/v1/rooms/{roomId}

Get Room by ID or External ID

Tags: [rooms](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
roomId	ID or External ID of Room to Get	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[Room](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## PUT /public/api/v1/rooms/{roomId}

Update Room by ID or External ID

Tags: [rooms](#)

### DESCRIPTION

### REQUEST BODY

## Room data

[CreateUpdateRoom](#)

### REQUEST PARAMETERS

Name	Description	Type	Data type	
roomId	ID or External ID of the Room to Update	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[Room](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## GET

Get Network Connection Protocol and Port for a given Room

**/public/api/v1/rooms/{roomId}/device-access**

Tags: [room-configuration](#)

### DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
roomId	ID of the Room	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

[DeviceAccess](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Not Found

### 500 Internal Server Error

Internal Error

## GET [/public/api/v1/rooms/{roomId}/hardware1/inputs](#)

Get Hardware 1 (SCHD) Input Configuration

Tags: [room-configuration](#)

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
roomId	ID of the Room	path	string	<b>required</b>

## RESPONSES

**200 OK**

Successful

[Hardware1Inputs](#)**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Not Found

**500 Internal Server Error**

Internal Error

**GET**

Get the enabled status of a Hardware 2 (Echo360 PRO) front panel

**/public/api/v1/rooms/{roomId}/hardware2/front-panel**Tags: [room-configuration](#)**DESCRIPTION****REQUEST PARAMETERS**

Name	Description	Type	Data type
roomId	ID of the Room	path	<i>string</i> <b>required</b>

**RESPONSES****200 OK**

Successful

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Not Found

**500 Internal Server Error**

Internal Error

**GET**

Get Hardware 2 (Echo360 PRO) Input Configuration

**/public/api/v1/rooms/{roomId}/hardware2/inputs**

Tags: [room-configuration](#)

**DESCRIPTION**

**REQUEST PARAMETERS**

Name	Description	Type	Data type	
roomId	ID of the Room	path	<i>string</i>	<b>required</b>

**RESPONSES**

**200 OK**

Successful

[Hardware2Inputs](#)

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Not Found

**500 Internal Server Error**

Internal Error

**GET** Get the One Touch details of a front-panel-enabled Hardware 2 (Echo360 PRO) device  
**/public/api/v1/rooms/{roomId}/hardware2/one-touch**  
Tags: [room-configuration](#)

#### DESCRIPTION

#### REQUEST PARAMETERS

Name	Description	Type	Data type	
roomId	ID of the Room	path	<i>string</i>	<b>required</b>

#### RESPONSES

**200 OK**  
Successful

[OneTouch](#)

**400 Bad Request**  
Bad Request

**401 Unauthorized**  
Not Authenticated

**403 Forbidden**  
Forbidden

**404 Not Found**  
Not Found

**500 Internal Server Error**  
Internal Error

**GET** Get the enabled status of a Hardware 3 (Echo360 POD) front panel  
**/public/api/v1/rooms/{roomId}/hardware3/front-panel**  
Tags: [room-configuration](#)

#### DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
roomId	ID of the Room	path	<i>string</i>	<b>required</b>

## RESPONSES

### 200 OK

Successful

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Not Found

### 500 Internal Server Error

Internal Error

## GET

Get Hardware 3 (Echo360 POD) Input Configuration

**/public/api/v1/rooms/{roomId}/hardware3/inputs**

Tags: [room-configuration](#)

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
roomId	ID of the Room	path	<i>string</i>	<b>required</b>

## RESPONSES

**200 OK**

Successful

[Hardware3Inputs](#)**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Not Found

**500 Internal Server Error**

Internal Error

**GET**

Get the One Touch details of a front-panel-enabled Hardware 3 (Echo360 POD) device

**/public/api/v1/rooms/{roomId}/hardware3/one-touch**Tags: [room-configuration](#)**DESCRIPTION****REQUEST PARAMETERS**

Name	Description	Type	Data type
roomId	ID of the Room	path	<i>string</i> <b>required</b>

**RESPONSES****200 OK**

Successful

[OneTouch](#)**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Not Found

**500 Internal Server Error**

Internal Error

**GET /public/api/v1/rooms/{roomId}/local-user/admin**

Get Local Admin User Credentials

Tags: [room-configuration](#)**DESCRIPTION****REQUEST PARAMETERS**

Name	Description	Type	Data type
roomId	ID of the Room	path	<i>string</i> <b>required</b>

**RESPONSES****200 OK**

Successful

[LocalUser](#)**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Not Found

**500 Internal Server Error**

Internal Error

## GET /public/api/v1/rooms/{roomId}/local-user/generic

Get Local Generic User Credentials

Tags: [room-configuration](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
roomId	ID of the Room	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[LocalUser](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Not Found

#### 500 Internal Server Error

Internal Error

## GET /public/api/v1/rooms/{roomId}/network

Get Network Settings plus Time Servers for a given Room

Tags: [room-configuration](#)

### DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
roomId	ID of the Room	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

[Network](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Not Found

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/rooms/{roomId}/outbound-proxy

Get Outbound Proxy details for a given Room

Tags: [room-configuration](#)

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
roomId	ID of the Room	path	string	<b>required</b>

## RESPONSES

**200 OK**

Successful

[ProxyOutbound](#)**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Not Found

**500 Internal Server Error**

Internal Error

**GET /public/api/v1/rooms/{roomId}/software1/inputs** Get Software (CCAP) Input ConfigurationTags: [room-configuration](#)**DESCRIPTION****REQUEST PARAMETERS**

Name	Description	Type	Data type
roomId	ID of the Room	path	<i>string</i> <b>required</b>

**RESPONSES****200 OK**

Successful

[SoftwareInputs](#)**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Not Found

**500 Internal Server Error**

Internal Error

**GET /public/api/v1/schedules**

List of all Schedules for current Institution

Tags: [schedules](#)

**DESCRIPTION**

**REQUEST PARAMETERS**

Name	Description	Type	Data type
roomId	ID or External ID of the Room to filter Schedules	query	<i>string</i> (default: "None")
sectionId	ID or External ID of the Section to filter Schedules	query	<i>string</i> (default: "None")
limit	Number of results	query	<i>integer</i> (int32)
offset	Key returned from prior result to get next page	query	<i>string</i>

**RESPONSES**

**200 OK**

Successful

**ITEMS**

[Schedule](#)

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**405 Method Not Allowed**

Method Not Allowed

**500 Internal Server Error**

Internal Error

**POST /public/api/v1/schedules**[Add Schedule](#)

Tags: [schedules](#)

**DESCRIPTION****REQUEST BODY**

Schedule data

```
CreateOrUpdateSchedule
```

**RESPONSES****200 OK**

Successful

```
Schedule
```

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**500 Internal Server Error**

Internal Error

## DELETE /public/api/v1/schedules/{scheduleId}

Delete Schedule by ID or External ID

Tags: [schedules](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
scheduleId	ID or External ID of the Schedule to Delete	path	string	<b>required</b>

### RESPONSES

#### 200 OK

Successful

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 500 Internal Server Error

Internal Error

## GET /public/api/v1/schedules/{scheduleId}

Get Schedule by ID or External ID

Tags: [schedules](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
scheduleId	ID or External ID of the Schedule to Get	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

[Schedule](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## PUT /public/api/v1/schedules/{scheduleId}

Tags: [schedules](#)

Update Schedule by ID or External ID

### DESCRIPTION

### REQUEST BODY

Schedule data

[CreateOrUpdateSchedule](#)

### REQUEST PARAMETERS

Name	Description	Type	Data type
scheduleId	ID or External ID of the Schedule to Update	path	string <span>required</span>

## RESPONSES

### 200 OK

Successful

[Schedule](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## PUT

Update Schedule's Device Options by ID or External ID

**/public/api/v1/schedules/{scheduleId}/deviceOptions**

Tags: [schedules](#)

### DESCRIPTION

### REQUEST BODY

Schedule Device Options data

[UpdateScheduleDeviceOptions](#)

### REQUEST PARAMETERS

Name	Description	Type	Data type
------	-------------	------	-----------

scheduleId ID or External ID of the Schedule to Update Device  
Options

path *string*

**required**

## RESPONSES

### 200 OK

Successful

[Schedule](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## PUT

Update Schedule's External ID by ID or External ID

**/public/api/v1/schedules/{scheduleId}/external\_id**

Tags: [schedules](#)

## DESCRIPTION

## REQUEST BODY

Schedule External ID

[UpdateExternalId](#)

## REQUEST PARAMETERS

Name	Description	Type	Data type
scheduleId	ID or External ID of the Schedule to Update External ID	path	string <span>required</span>

## RESPONSES

### 200 OK

Successful

[Schedule](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

**PUT /public/api/v1/schedules/{scheduleId}/flags** Update Schedule's Flags by ID or External ID  
Tags: [schedules](#)

## DESCRIPTION

## REQUEST BODY

Schedule flags data

[UpdateScheduleFlags](#)

## REQUEST PARAMETERS

Name	Description	Type	Data type	
scheduleId	ID or External ID of the Schedule to Update Flags	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

[Schedule](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## PUT

Update Schedule's Instructors by ID or External ID

**/public/api/v1/schedules/{scheduleId}/instructors**

Tags: [schedules](#)

## DESCRIPTION

## REQUEST BODY

Schedule Instructors

[UpdateScheduleInstructors](#)

## REQUEST PARAMETERS

Name	Description	Type	Data type	
scheduleId	ID or External ID of the Schedule to Update Instructors	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

[Schedule](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

**PUT /public/api/v1/schedules/{scheduleId}/name** Update Schedule's Name by ID or External ID  
Tags: [schedules](#)

## DESCRIPTION

## REQUEST BODY

Schedule Name data

[UpdateScheduleName](#)

## REQUEST PARAMETERS

Name	Description	Type	Data type	
scheduleId	ID or External ID of the Schedule to Update Name	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

[Schedule](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## PUT

**/public/api/v1/schedules/{scheduleId}/section**

Update Schedule's Section by ID or External ID

Tags: [schedules](#)

## DESCRIPTION

## REQUEST BODY

Schedule Section data

[UpdateScheduleSection](#)

## REQUEST PARAMETERS

Name	Description	Type	Data type	
scheduleId	ID or External ID of the Schedule to Update Section	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

[Schedule](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

**PUT /public/api/v1/schedules/{scheduleId}/times** Update Schedule's Times by ID or External ID

Tags: [schedules](#)

## DESCRIPTION

## REQUEST BODY

Schedule Time data

[UpdateScheduleTimes](#)

## REQUEST PARAMETERS

Name	Description	Type	Data type	
scheduleId	ID or External ID of the Schedule to Update Times	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

[Schedule](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/sections

List of all Sections for current Institution

Tags: [sections](#)

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type
courseId	ID or External ID of the Course to filter Sections	query	string (default: "None")

termId	ID or External ID of the Term to filter Sections	query	string (default: "None")
limit	Number of results	query	integer (int32)
offset	Key returned from prior result to get next page	query	string

## RESPONSES

### 200 OK

Successful

#### ITEMS

[Section](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## POST /public/api/v1/sections

Add Section

Tags: [sections](#)

### DESCRIPTION

### REQUEST BODY

Section data

[CreateSection](#)

## RESPONSES

### 200 OK

Successful

[Section](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 500 Internal Server Error

Internal Error

## DELETE /public/api/v1/sections/{sectionId}

Delete Section by ID or External ID

Tags: [sections](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Delete	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/sections/{sectionId}

Get Section by ID or External ID

Tags: [sections](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Get	path	string	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[Section](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## PUT /public/api/v1/sections/{sectionId}

Update Section by ID or External ID

Tags: [sections](#)

### DESCRIPTION

### REQUEST BODY

Section data

[UpdateSection](#)

### REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Update	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[Section](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## POST /public/api/v1/sections/{sectionId}/clone

Tags: [sections](#)

Clone Section by ID or External ID

### DESCRIPTION

### REQUEST BODY

Section data

[CloneSection](#)

### REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Clone	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[Section](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## GET

List of all existing and future Lessons for current Section

**/public/api/v1/sections/{sectionId}/lessons**

Tags: [lessons](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Get Lessons	path	string	<b>required</b>

### RESPONSES

#### 200 OK

Successful

#### ITEMS

[Lesson](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

**POST /public/api/v1/sections/{sectionId}/lessons**

Add Lesson

Tags: [lessons](#)

## DESCRIPTION

## REQUEST BODY

Lesson data

Lesson

## REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Create Lesson	path	<i>string</i>	<b>required</b>

## RESPONSES

### 200 OK

Successful

Lesson

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 500 Internal Server Error

Internal Error

**PUT**  
**/public/api/v1/sections/{sectionId}/lms-**  
**course-ids**

Tags: [sections](#)

Update Section LMS Course IDs by ID or External ID

## DESCRIPTION

## REQUEST BODY

Section data

```
UpdateSectionLmsCourses
```

## REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Update LMS Course IDs	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

```
Section
```

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/sections/{sectionId}/lti-links

Tags: [lti-links](#)

List all LTI Links for a single section

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Get LTI Links	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

#### ITEMS

[LtiLink](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 500 Internal Server Error

Internal Error

## PUT

Update Section Secondary Instructors by ID or External ID

**/public/api/v1/sections/{sectionId}/secondary-instructors**

Tags: [sections](#)

## DESCRIPTION

## REQUEST BODY

Section data

[UpdateSectionInstructors](#)

## REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Update Secondary Instructors	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

[Section](#)

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/sections/{sectionId}/users

List of all User enrollments for Section

Tags: [enrollment](#)

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Get Enrollments	path	string	<b>required</b>

## RESPONSES

### 200 OK

Successful

#### ITEMS

*UserEnrollment*

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 500 Internal Server Error

Internal Error

## POST /public/api/v1/sections/{sectionId}/users

Enroll User in a Section

Tags: [enrollment](#)

### DESCRIPTION

### REQUEST BODY

Section data

[EnrollUser](#)

### REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Enroll User	path	<i>string</i>	<b>required</b>

## RESPONSES

### 200 OK

Successful

UserEnrollment

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**500 Internal Server Error**

Internal Error

**DELETE /public/api/v1/sections/{sectionId}/users/{userId}**

Unenroll User from Section

Tags: [enrollment](#)

**DESCRIPTION**

**REQUEST PARAMETERS**

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Delete Enrollment	path	string	<b>required</b>
userId	ID or External ID of the User to Delete Enrollment	path	string	<b>required</b>

**RESPONSES**

**200 OK**

Successful

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**500 Internal Server Error**

Internal Error

## GET /public/api/v1/sections/{sectionId}/users/{userId}

Get User enrollment for Section

Tags: [enrollment](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID or External ID of the Section to Get Enrollments	path	<i>string</i>	<b>required</b>
userId	ID or External ID of the User to Get Enrollments	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[UserEnrollment](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 500 Internal Server Error

Internal Error

## PUT /public/api/v1/sections/{sectionId}/users/{userId}

Update a User's Section Role

Tags: [enrollment](#)

### DESCRIPTION

### REQUEST BODY

Either "Student" or "Instructor"

## REQUEST PARAMETERS

Name	Description	Type	Data type	
sectionId	ID of the Section to Update Section Role	path	string	required
userId	ID of the User to Update Section Role	path	string	required

## RESPONSES

### 200 OK

Successful

UserEnrollment

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/terms

Tags: [terms](#)

List of all Terms for current Institution

## DESCRIPTION

## REQUEST PARAMETERS

Name	Description	Type	Data type
limit	Number of results	query	integer (int32)

offset    Key returned from prior result to get next page

query    *string*

## RESPONSES

### 200 OK

Successful

#### ITEMS

*Term*

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## POST /public/api/v1/terms

Add Term

Tags: [terms](#)

### DESCRIPTION

### REQUEST BODY

Term data

[CreateUpdateTerm](#)

## RESPONSES

**200 OK**

Successful

Term

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**500 Internal Server Error**

Internal Error

**DELETE /public/api/v1/terms/{termId}**

Delete Term by ID or External ID

Tags: [terms](#)**DESCRIPTION****REQUEST PARAMETERS**

Name	Description	Type	Data type
termId	ID or External ID of the Term to Delete	path	string <span>required</span>

**RESPONSES****200 OK**

Successful

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

### 404 Not Found

Resource Not Found

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/terms/{termId}

Get Term by ID or External ID

Tags: [terms](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
termId	ID or External ID of the Term to Get	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

[Term](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## PUT /public/api/v1/terms/{termId}

Update Term by ID or External ID

Tags: [terms](#)

### DESCRIPTION

### REQUEST BODY

Term data

```
CreateUpdateTerm
```

### REQUEST PARAMETERS

Name	Description	Type	Data type	
termId	ID or External ID of the Term to Update	path	<i>string</i>	<b>required</b>

### RESPONSES

#### 200 OK

Successful

```
Term
```

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## GET /public/api/v1/users

List of all Users for current Institution

Tags: [users](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type
limit	Number of results	query	<i>integer (int32)</i>
offset	Key returned from prior result to get next page	query	<i>string</i>

### RESPONSES

#### 200 OK

Successful

#### ITEMS

[User](#)

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

#### 500 Internal Server Error

Internal Error

## POST /public/api/v1/users

Add User

Tags: [users](#)

### DESCRIPTION

## REQUEST BODY

User data

CreateUpdateUser

## RESPONSES

### 200 OK

Successful

User

### 400 Bad Request

Bad Request

### 401 Unauthorized

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/users/{userId}

Get User by ID, Email, or External ID

Tags: [users](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
userId	ID, Email, or External ID of the User to Get	path	string	<b>required</b>

### RESPONSES

## 200 OK

Successful

User

## 400 Bad Request

Bad Request

## 401 Unauthorized

Not Authenticated

## 403 Forbidden

Forbidden

## 404 Not Found

Resource Not Found

## 405 Method Not Allowed

Method Not Allowed

## 500 Internal Server Error

Internal Error

## PUT /public/api/v1/users/{userId}

Update User by ID, Email, or External ID

Tags: [users](#)

### DESCRIPTION

### REQUEST BODY

User data

CreateUpdateUser

### REQUEST PARAMETERS

Name	Description	Type	Data type	
userId	ID, Email, or External ID of the User to Update	path	string	<b>required</b>

### RESPONSES

**200 OK**

Successful

User

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

**403 Forbidden**

Forbidden

**404 Not Found**

Resource Not Found

**405 Method Not Allowed**

Method Not Allowed

**500 Internal Server Error**

Internal Error

**GET /public/api/v1/users/{userId}/invite**

Resend User Invite by ID, Email, or External ID

Tags: [users](#)**DESCRIPTION****REQUEST PARAMETERS**

Name	Description	Type	Data type
userId	ID, Email, or External ID of the User to resend invite	path	string <b>required</b>

**RESPONSES****200 OK**

Successful

**400 Bad Request**

Bad Request

**401 Unauthorized**

Not Authenticated

### 403 Forbidden

Forbidden

### 404 Not Found

Resource Not Found

### 405 Method Not Allowed

Method Not Allowed

### 500 Internal Server Error

Internal Error

## GET /public/api/v1/users/{userId}/status/{status}

Disable / Enable User by ID, Email, or External ID

Tags: [users](#)

### DESCRIPTION

### REQUEST PARAMETERS

Name	Description	Type	Data type	
userId	ID, Email, or External ID of the User to Disable / Enable	path	string	required
status	Active or Inactive status of the User to Disable / Enable	path	string	required

### RESPONSES

#### 200 OK

Successful

#### 400 Bad Request

Bad Request

#### 401 Unauthorized

Not Authenticated

#### 403 Forbidden

Forbidden

#### 404 Not Found

Resource Not Found

#### 405 Method Not Allowed

Method Not Allowed

## 500 Internal Server Error

Internal Error

# Schema definitions

ActionAnyContent: *object*

Availability: *object*

### PROPERTIES

**available:** *boolean* **required**

Availability

**when:** *string*

Date when media will be available: YYYY-MM-DD

Building: *object*

### PROPERTIES

**id:** *string* **required**

Unique Identifier of the Building

**campusId:** *string* **required**

ID of the Campus to which the Building belongs

**institutionId:** *string* **required**

ID of the Institution to which the Campus belongs

**name:** *string* **required**

Name of the Building returned

**externalId:** *string*

External system identifier for the current Building

Campus: *object*

## PROPERTIES

**id:** *string* **required**

Unique Identifier of the Campus

**institutionId:** *string* **required**

ID of the Institution to which the Campus belongs

**name:** *string* **required**

Name of the Campus returned

**timeZone:** *string* **required**

The Time Zone in which the Campus is located

### EXAMPLE

```
"US/Eastern"
```

**timeZoneOffsetMinutes:** *integer (int32)* **required**

Offset in minutes to add to UTC to get the local time.

### EXAMPLE

```
-120
```

**externalId:** *string*

External system identifier for the current Campus

Chronology: *object*

## PROPERTIES

**zone:** *DateTimeZone*

CloneSection: *object*

## PROPERTIES

**courseId:** *string* **required**

ID or External ID of the Course to which the cloned Section belongs

### EXAMPLE

```
"ICT102"
```

**termId:** *string* **required**

ID or External ID of the Term to which the cloned Section belongs

**EXAMPLE**

```
"SEM1_2017"
```

**sectionNumber:** *string* **required**

Number to uniquely identify the cloned Section

**EXAMPLE**

```
"ICT102_S1_2017"
```

**description:** *string*

Optional Description of the cloned Section

**EXAMPLE**

```
"Cloned Section"
```

**instructorId:** *string*

ID, Email, or External ID of the Instructor for the cloned Section (if omitted the existing Section Instructor will be kept)

**EXAMPLE**

```
"Foo.Bar@echo360.com"
```

**externalId:** *string*

External system identifier of the cloned Section

**EXAMPLE**

```
"ICT102_S2_2017"
```

**ContentAvailability:** *object*

Course: *object*

#### PROPERTIES

**id:** *string* **required**

Unique Identifier of the Course

**institutionId:** *string* **required**

ID of the Instituion to which the Organization belongs

**organizationId:** *string*

ID of the Organization to which the Department belongs. Will default to ownership by Institution if non-existent

**departmentId:** *string*

ID of the Department to which the Course belongs. Will default to ownership by Organization if non-existent

**name:** *string* **required**

Name of the Course returned

**courseIdentifier:** *string* **required**

Institution-specific identifier for the Course

#### EXAMPLE

"ICT102"

**sectionCount:** *integer (int64)* **required**

Number of Sections that exist for the current Course

#### EXAMPLE

4

**externalId:** *string*

External system identifier

CreateOrUpdateSchedule: *object*

## PROPERTIES

**startDate:** *string* **required**

Start date for capture in ISO 8601 Date format: YYYY-MM-DD

### EXAMPLE

```
"2017-02-01"
```

**startTime:** *string* **required**

Start time for the capture (to the minute) in ISO 8601 Time format: HH:MM. The time zone for the schedule's linked room will be used

### EXAMPLE

```
"09:00"
```

**endDate:** *string*

The last date for the capture in ISO 8601 Date format: YYYY-MM-DD. If this is specified, `daysOfWeek` must also be provided

### EXAMPLE

```
"2017-05-31"
```

**daysOfWeek:** *string[]*

Days of the week this schedule should capture as 2 letter abbreviations: SU, MO, TU, WE, TH, FR, SA. This is required when `endDate` is specified

### EXAMPLE

```
"[\\"MO\\", \\"TU\\"]"
```

## ITEMS

*string*

**exclusionDates:** *string[]*

Dates within the scheduled period skip recording in ISO 8601 Date format: YYYY-MM-DD. This is unused when no `endDate/daysOfWeek` is specified

### EXAMPLE

```
"[\\"2017-04-14\\", \\"2014-04-17\\"]"
```

## ITEMS

*string*

**durationMinutes:** *integer (int32)* **required**

Duration of each scheduled capture in Minutes

## EXAMPLE

```
60
```

**termId:** *string*

ID of the Term in which to Schedule the Capture

**sectionId:** *string*

Deprecated -- ID or External ID of the Section in which to Schedule the Capture

## EXAMPLE

```
"ICT102_S1_2017"
```

**sections:** *object[]*

List of the Sections in which to Schedule the Capture

## ITEMS

*ScheduleSection*

**name:** *string*

Name of the Schedule

## EXAMPLE

```
"ICT102 Sem1 2017 9:00-10:00 Mon/Tue, Rm: Rockerfeller Studio 6H"
```

**roomId:** *string* **required**

ID or External ID of the Room in which to Schedule the Capture

## EXAMPLE

```
"ROCKEFELLER_STUDIO_6H"
```

**instructorId:** *string*

ID, Email, or External ID of the Instructor

**EXAMPLE**

```
"Foo.Bar@echo360.com"
```

**guestInstructor:** *string*

Name of any Guest Instructor

**EXAMPLE**

```
"Visiting consultant Prof A. Citizen"
```

**shouldCaption:** *boolean* **required**

Boolean value indicating the Capture should be Captioned

**shouldStreamLive:** *boolean* **required**

Boolean value indicating the Capture should be Streamed Live

**input1:** *string*, x { "display", "video", "altvideo" }

Input 1 Logical Graphics Type

**input2:** *string*, x { "display", "video", "altvideo" }

Input 2 Logical Graphics Type

**captureQuality:** *string*, x { "medium", "high", "highest" } **required**

Quality of the Capture

**streamQuality:** *string*, x { "medium", "high", "highest" }

Streaming Quality of the Capture

**externalId:** *string*

External system identifiers of the Schedule

**EXAMPLE**

```
"ICT102S12017-RECUR_MON,TUE_0900-1000_ROCKEFELLER_STUDIO_6H"
```

**resolvedSections:** *object[]*

**ITEMS**

[ScheduleSection](#)

## CreateSection: *object*

### PROPERTIES

**courseId:** *string* **required**

ID or External ID of the Course to which the Section belongs

#### EXAMPLE

```
"ICT102"
```

**termId:** *string* **required**

ID or External ID of the Term to which the Section belongs

#### EXAMPLE

```
"SEM1_2017"
```

**sectionNumber:** *string* **required**

Number to uniquely identify each Section

#### EXAMPLE

```
"ICT102_S1_2017"
```

**description:** *string*

Optional Description of the current Section

**instructorId:** *string*

ID, Email, or External ID of the Instructor by whom the Section will be taught

#### EXAMPLE

```
"Foo.Bar@echo360.com"
```

**secondaryInstructorIds:** *string[]*

Array of ID's of the Secondary Instructors for the current Section

#### ITEMS

*string*

**ImCourseIds:** *string[]*

Array of Section Identifiers from any External Learning Management System

**EXAMPLE**

```
"[\"ICT102_S1_2017\"]"
```

**ITEMS**

*string*

**ImCourses:** *string[]*

Array of Section Identifiers from any External Learning Management System

**ITEMS**

*string*

**externalId:** *string*

External system identifier of the Section

**EXAMPLE**

```
"ICT102_S1_2017"
```

**CreateUpdateBuilding:** *object*

**PROPERTIES**

**campusId:** *string* **required**

ID or External of the Campus to which the Building belongs

**EXAMPLE**

```
"NYC"
```

**name:** *string* **required**

Name of the Building

**EXAMPLE**

```
"Rockefeller Plaza"
```

**externalId:** *string*

External system identifier of the Building

## EXAMPLE

"ROCKEFELLER"

## CreateUpdateCampus: *object*

### PROPERTIES

**name:** *string* **required**

Name of the Campus

#### EXAMPLE

"New York City"

**timeZone:** *string*

The Time Zone in which the Campus is located. Will default to the Institution's Time Zone if not specified

#### EXAMPLE

"US/Eastern"

**externalId:** *string*

External system identifier of the Campus

#### EXAMPLE

"NYC"

## CreateUpdateCourse: *object*

### PROPERTIES

**organizationId:** *string*

ID or External ID of the Organization to which the Department belongs. Will default to ownership by Institution if left empty.

#### EXAMPLE

"FAC\_ENGINEERING"

**departmentId:** *string*

ID or External ID of the Department to which the Course belongs. Will default to ownership by Organization if left empty

**EXAMPLE**

```
"DEPT_CSEE"
```

**name:** *string* **required**

Name of the Course

**EXAMPLE**

```
"Introduction to Computer Science"
```

**courseIdentifier:** *string* **required**

Institution-specific identifier for the Course

**EXAMPLE**

```
"ICT102"
```

**externalId:** *string*

External system identifier of the Course

**EXAMPLE**

```
"ICT102"
```

**CreateUpdateDepartment:** *object*

**PROPERTIES**

**organizationId:** *string*

The ID or External ID of the Organization to which the Department belongs. Will default to ownership by Institution if left empty.

**EXAMPLE**

```
"FAC_ENGINEERING"
```

**name:** *string* **required**

Name of the Department

**EXAMPLE**

```
"Dept. of Computer Science & Electrical Engineering"
```

**externalId:** *string*

External system identifier of the Department

**EXAMPLE**

```
"DEPT_CSEE"
```

CreateUpdateOrganization: *object*

**PROPERTIES**

**name:** *string* **required**

Name of the Organization

**EXAMPLE**

```
"Faculty of Engineering"
```

**externalId:** *string*

External system identifier of the Organization

**EXAMPLE**

```
"FAC_ENGINEERING"
```

CreateUpdateRoom: *object*

**PROPERTIES**

**buildingId:** *string* **required**

ID or External ID of the Building to which to the Room belongs

### EXAMPLE

```
"ROCKEFELLER"
```

**name:** *string* **required**

Name of the Room

### EXAMPLE

```
"Rockerfeller Studio 6H"
```

**deviceId:** *string*

Device ID to assign this room

**externalId:** *string*

External system identifier of the Room

### EXAMPLE

```
"ROCKEFELLER_STUDIO_6H"
```

CreateUpdateTerm: *object*

### PROPERTIES

**name:** *string* **required**

Name of the Term

### EXAMPLE

```
"Semester 1 2017"
```

**session:** *DateRange* **required**

Object consisting of the Start Date and End Date of the Term

**exceptions:** *object[]* **required**

Any Dates within the current Term on which Class will not be held

### ITEMS

*DateRange*

**externalId:** *string*

External system identifier of the Term

**EXAMPLE**

```
"SEM1_2017"
```

**CreateUpdateUser:** *object*

**PROPERTIES**

**email:** *string* **required**

Email address of the User

**EXAMPLE**

```
"Foo.Bar@echo360.com"
```

**timeZone:** *string*

The Time Zone in which the User is located

**EXAMPLE**

```
"US/Eastern"
```

**firstName:** *string* **required**

First Name of the User

**EXAMPLE**

```
"Foo"
```

**lastName:** *string* **required**

Last Name of the User

**EXAMPLE**

```
"Bar"
```

**phoneNumber:** *PhoneNumber*

Phone Number of the User. An object containing the Region and Number

#### EXAMPLE

```
{"region": "US", "number": "000-867-5309"}
```

**profileImageUrl:** *string*

URL of the User's Profile Image

**roles:** *string[]* **required**

Role(s) of the User within the Institution: Admin, Instructor, Student

#### EXAMPLE

```
["Admin", "Instructor", "Student"]
```

#### ITEMS

*string*

**ssold:** *string*

A string that identifies the user to an external identity provider

#### EXAMPLE

```
fbar+internet2.edu@subdomain.incommon.org
```

**externalId:** *string*

External system identifier of the User

#### EXAMPLE

```
Foo.Bar@echo360.com
```

**DateRange:** *object*

#### PROPERTIES

**startDate:** *string (date)* **required**

Start Date of the current Term in ISO 8601 Date format: YYYY-MM-DD

**endDate:** *string (date)* **required**

End Date of the current Term in ISO 8601 Date format: YYYY-MM-DD

**DateTimeField:** *object*

**PROPERTIES**

**lenient:** *boolean*

**durationField:** *DurationField*

**minimumValue:** *integer (int32)*

**maximumValue:** *integer (int32)*

**rangeDurationField:** *DurationField*

**leapDurationField:** *DurationField*

**name:** *string*

**type:** *DateTimeFieldType*

**supported:** *boolean*

**DateTimeFieldType:** *object*

**PROPERTIES**

**durationType:** *DurationFieldType*

**rangeDurationType:** *DurationFieldType*

**name:** *string*

**DateTimeRange:** *object*

**PROPERTIES**

**start:** *LocalDateTime*

**end:** *LocalDateTime*

**validations:** *ValidationGroupDateTimeRange*

**date:** *string*

**startTime:** *string*

**endTime:** *string*

**dateName:** *string*

**shortDate:** *string*

**shortDateWithWeekDay:** *string*

**smallDateWithWeekDay:** *string*

**DateTimeZone:** *object*

#### PROPERTIES

**fixed:** *boolean*

**id:** *string*

**Department:** *object*

#### PROPERTIES

**id:** *string* **required**

Unique Identifier of the Department

**organizationId:** *string*

The ID of the Organization to which the Department belongs. Will default to ownership by Institution if non-existent.

**institutionId:** *string* **required**

ID of the Institution to which the Organization belongs

**name:** *string* **required**

Name of the Department returned

**courseCount:** *string* **required**

Number of Courses that exist for the current Department

#### EXAMPLE

"10"

**sectionCount:** *string* **required**

Number of Sections that exist for the current Department

#### EXAMPLE

"4"

**externalId:** *string*

External system identifier for the current Department

**DeviceAccess:** *object*

#### PROPERTIES

**protocol:** *string*, x { "http", "https" } **required**

Protocol

**port:** *integer (int32)* **required**

Port number

**DurationField:** *object*

#### PROPERTIES

**unitMillis:** *integer (int64)*

**precise:** *boolean*

**name:** *string*

**type:** *DurationFieldType*

**supported:** *boolean*

**DurationFieldType:** *object*

#### PROPERTIES

**name:** *string*

**EnrollUser:** *object*

#### PROPERTIES

**userId:** *string* **required**

ID or Email of the User

**role:** *string*, x { "Student", "Instructor" } **required**

Role(s) of the User within the Institution

## Hardware1Inputs: *object*

### PROPERTIES

**audioInput:** *string*, x { "rca", "balanced", "line-in", "mic", "none" }

Audio Input Connector

**audioAnalogGain:** *integer (int32)*

Gain value for analog Audio

**graphics1DviEnabled:** *string*

Enabled status of DVI Input

**graphics1DviGraphicsInput:** *string*, x { "composite", "dvi", "dvi\_a", "dvi\_d", "hdmi", "sdi", "vga" }

Connector for DVI Graphics Input

**graphics1DviAspectRatio:** *string*, x { "4x3", "16x9" }

Aspect ratio of the DVI input

**graphics1DviSourceType:** *string*, x { "video", "display" }

Type of the DVI Input Source

**graphics1CompositeEnabled:** *boolean*

Enabled status of Composite Input

**graphics1CompositeStandard:** *string*, x { "ntsc", "pal" }

Video Standard of the Composite Input

**graphics2DviEnabled:** *string*

Enabled status of DVI Input

**graphics2DviGraphicsInput:** *string*, x { "composite", "dvi", "dvi\_a", "dvi\_d", "hdmi", "sdi", "vga" }

Connector for DVI Graphics Input

**graphics2DviAspectRatio:** *string*, x { "4x3", "16x9" }

Aspect ratio of the DVI input

**graphics2DviSourceType:** *string*, x { "video", "display" }

Type of the DVI Input Source

**graphics2CompositeEnabled:** *boolean*

Enabled status of Composite Input

**graphics2CompositeStandard:** *string*, x { "ntsc", "pal" }  
Video Standard of the Composite Input

## Hardware2Inputs: *object*

### PROPERTIES

**audioInput:** *string*, x { "rca", "balanced", "line-in", "mic", "none" }  
Audio Input Connector

**audioAnalogGain:** *integer (int32)*  
Gain value for analog Audio

**graphics1CompositeEnabled:** *boolean*  
Enabled status of Composite Input

**graphics1CompositeSourceType:** *string*, x { "video", "display" }  
Type of the Composite Input Source

**graphics1CompositeStandard:** *string*, x { "ntsc", "pal" }  
Video Standard of the Composite Input

**graphics1HdmiEnabled:** *boolean*  
Enabled status of the HDMI input

**graphics1HdmiSourceType:** *string*, x { "video", "display" }  
Type of the HDMI Input Source

**graphics1HdmiAspectRatio:** *string*, x { "4x3", "16x9" }  
Aspect ratio of the HDMI input

**graphics1HdmiMixAudio:** *boolean*  
Enabled status of HDMI Mixed Audio

**graphics1SdiEnabled:** *boolean*  
Enabled status of SDI Input

**graphics1SdiSourceType:** *string*, x { "video", "display" }  
Type of the SDI Input Source

**graphics1SdiAspectRatio:** *string*, x { "4x3", "16x9" }  
Aspect ratio of the SDI input

**graphics1VgaEnabled:** *boolean*

Enabled status of VGA Input

**graphics1VgaSourceType:** *string*, x { "video", "display" }

Type of the VGA Input Source

**graphics1VgaAspectRatio:** *string*, x { "4x3", "16x9" }

Aspect ratio of the VGA input

**graphics2CompositeEnabled:** *boolean*

Enabled status of Composite Input

**graphics2CompositeStandard:** *string*, x { "ntsc", "pal" }

Video Standard of the Composite Input

**graphics2CompositeSourceType:** *string*, x { "video", "display" }

Type of the Composite Input Source

**graphics2HdmiEnabled:** *boolean*

Enabled status of HDMI Input

**graphics2HdmiSourceType:** *string*, x { "video", "display" }

Type of the HDMI Input Source

**graphics2HdmiAspectRatio:** *string*, x { "4x3", "16x9" }

Aspect ratio of the HDMI input

**graphics2HdmiMixAudio:** *boolean*

Enabled status of HDMI Mixed Audio

**graphics2SdiEnabled:** *boolean*

Enabled status of SDI Input

**graphics2SdiSourceType:** *string*, x { "video", "display" }

Type of the SDI Input Source

**graphics2SdiAspectRatio:** *string*, x { "4x3", "16x9" }

Aspect ratio of the SDI input

**graphics2VgaEnabled:** *boolean*

Enabled status of VGA Input

**graphics2VgaSourceType:** *string*, x { "video", "display" }

Type of the VGA Input Source

**graphics2VgaAspectRatio:** *string*, x { "4x3", "16x9" }

Aspect ratio of the VGA input

**Hardware3Inputs:** *object*

#### PROPERTIES

**audioInput:** *string*, x { "rca", "balanced", "line-in", "mic", "none" }

Audio Input Connector

**audioAnalogGain:** *integer (int32)*

Gain value for analog Audio

**graphics1HdmiEnabled:** *boolean*

Enabled status of the HDMI Input

**graphics1HdmiAspectRatio:** *string*, x { "4x3", "16x9" }

Aspect ratio of the HDMI Input

**graphics1HdmiSourceType:** *string*, x { "video", "display" }

Type of the HDMI Input Source

**graphics1HdmiMixAudio:** *boolean*

Enabled status of HDMI Mixed Audio

**graphics2HdmiEnabled:** *boolean*

Enabled status of the HDMI Input

**graphics2HdmiAspectRatio:** *string*, x { "4x3", "16x9" }

Aspect ration of the HDMI Input

**graphics2HdmiSourceType:** *string*, x { "video", "display" }

Type of the HDMI Source

**graphics2HdmiMixAudio:** *boolean*

Enabled status of HDMI Mixed Audio

**Instant:** *object*

#### PROPERTIES

**millis:** *integer (int64)*

**chronology:** *Chronology*

**zone:** *DateTimeZone*

**afterNow:** *boolean*

**beforeNow:** *boolean*

**equalNow:** *boolean*

Lesson: *object*

## PROPERTIES

**id:** *string* **required**

Id of the lesson

**institutionId:** *string* **required**

Id of the institution

**sectionId:** *string* **required**

Id of the section

**captureOccurrenceId:** *string*

Id of the capture occurrence

**name:** *string*

Name of the lesson

**info:** *string*

Info of the lesson

**timing:** *DateTimeRange*

Timing of the lesson

**timeZone:** *string*

Name of the Time Zone

**shouldStreamLive:** *boolean*

Should stream live

**sharingLink:** *string*

Sharing link

**createdAt:** *Instant* **required**

DateTime value for Creation Date

## EXAMPLE

```
"2015-10-05T15:24:53.496Z"
```

**updatedAt:** *Instant* **required**

DateTime value for Updated Date

## EXAMPLE

```
"2015-10-05T15:24:53.496Z"
```

**LessonInfo:** *object*

### PROPERTIES

**name:** *string*

**info:** *string*

**timing:** *DateTimeRange*

**timeZone:** *string*

**validations:** *ValidationGroupLessonInfo*

**LessonMedia:** *object*

### PROPERTIES

**lessonId:** *string* **required**

Id of the lesson

**publishableMediaId:** *string* **required**

Id of the publishable media

**institutionId:** *string* **required**

Id of the institution

**termId:** *string*

Id of the term

**termName:** *string*

Name of the term

**courseId:** *string*

Id of the course

**sectionId:** *string* **required**

Id of the section

**lessonName:** *string*

Name of the lesson

**lessonTiming:** *DateTimeRange*

Time range for the lesson publishable media

**timeZone:** *string* **required**

Name of the timeZone

**courseName:** *string*

Name of the course

**contentAvailability:** *ContentAvailability* **required**

Availability of the content

**mediaType:** *string*, x { "presentation", "video" } **required**

Type of the media

**LmsProfile:** *object*

#### PROPERTIES

**id:** *string* **required**

Unique Identifier of the LmsProfile

**lmsName:** *string* **required**

Name of the Lms

**token:** *Token* **required**

Unique Token for LmsProfile

**lmsCartridge:** *string*

Lms Cartridge for LmsProfile

**courseIdField:** *string*

External System Id Ref to Course

**ItiEndpoint:** *string*

The Host Url for the LmsProfile

**LmsProfileInfo:** *object*

**PROPERTIES**

**ImsName:** *string* **required**

Name of the Lms

**courseIdField:** *string*

External System Id Ref to Course

**LocalDateTime:** *object*

**PROPERTIES**

**year:** *integer (int32)*

**dayOfMonth:** *integer (int32)*

**dayOfWeek:** *integer (int32)*

**era:** *integer (int32)*

**dayOfYear:** *integer (int32)*

**monthOfYear:** *integer (int32)*

**chronology:** [Chronology](#)

**millisOfDay:** *integer (int32)*

**hourOfDay:** *integer (int32)*

**minuteOfHour:** *integer (int32)*

**secondOfMinute:** *integer (int32)*

**millisOfSecond:** *integer (int32)*

**weekyear:** *integer (int32)*

**weekOfWeekyear:** *integer (int32)*

**yearOfEra:** *integer (int32)*

**yearOfCentury:** *integer (int32)*

**centuryOfEra:** *integer (int32)*

**values:** *integer[]*

**ITEMS**

*integer* (int32)

**fieldTypes:** *object[]*

**ITEMS**

[DateTimeFieldType](#)

**fields:** *object[]*

**ITEMS**

[DateTimeField](#)

**LocalUser:** *object*

**PROPERTIES**

**username:** *string* **required**

Local User credentials username

**password:** *string* **required**

Local User credentials password

**LtiLink:** *object*

**PROPERTIES**

**resourceLinkId:** *string* **required**

Resource Link ID from LMS: resource\_link\_id

**resourceLinkTitle:** *string*

Resource Link Title from LMS: resource\_link\_title

**lmsProfileId:** *string* **required**

ID of Echo360 LMS Profile

**contextId:** *string* **required**

Context ID from LMS: context\_id

**contextTitle:** *string*

Context Title from LMS: context\_title

**ImsCourseId:** *string*

Section Identifier from LMS

**objectId:** *string*

ID of Echo360 object (lesson or section) link is connected to

**objectType:** *string*

Type of Link: Section or Lesson

**institutionId:** *string* **required**

ID of Institution

**sectionId:** *string*

ID of Section link is connected to

**lessonId:** *string*

ID of Section link is connected to

**lisOutcomeServiceUrl:** *string*

Outcome Service Url from LMS: lis\_outcome\_service\_url

**gradebookMetric:** *string* **required**

Resource Link ID from LMS: resource\_link\_id

**LtiLinkInfo:** *object*

#### PROPERTIES

**resourceLinkId:** *string* **required**

Resource Link ID from LMS: resource\_link\_id

**resourceLinkTitle:** *string*

Resource Link Title from LMS: resource\_link\_title

**imsProfileId:** *string* **required**

ID of Echo360 LMS Profile

**contextId:** *string* **required**

Context ID from LMS: context\_id

**objectId:** *string*

ID of Echo360 object (lesson or section) link is connected to

**objectType:** *string*

Type of Link: Section or Lesson

**lisOutcomeServiceUrl:** *string*

Outcome Service Url from LMS: lis\_outcome\_service\_url

**gradebookMetric:** *string* **required**

Resource Link ID from LMS: resource\_link\_id

**validations:** [ValidationGroupLtiLinkInfo](#)

**Network:** *object*

### PROPERTIES

**networkType:** *string*, x { "DHCP" , "IPv4" } **required**

Type of the Network

**ipAddress:** *string*

IP Address of the Device on the Network

**subnetMask:** *string*

32-bit number to mask the IP address

**defaultGateway:** *string*

Node on the Network that can forward packets to other Networks

**primaryDns:** *string*

Primary Domain Name System of the Network

**secondaryDns:** *string*

Secondary Domain Name System of the Network

**primaryTimeServer:** *string* **required**

Primary Time Server of the Network

**secondaryTimeServer:** *string* **required**

Secondary Time Server of the Network

**wifiEnabled:** *boolean* **required**

WiFi Enabled flag

**wifiSsid:** *string*

WiFi SSID Name

**wifiPassword:** *string*

WiFi Password

## OneTouch: *object*

### PROPERTIES

**deviceType:** *string* **required**

The Type of Capture Device

**input1:** *string*, x { "display", "video", "altvideo" } **required**

The Type of the First Input to the One Touch Enabled Capture Device

**input2:** *string*, x { "display", "video", "altvideo" } **required**

The Type of the Second Input to the One Touch Enabled Capture Device

**quality:** *string*, x { "medium", "high" } **required**

The Quality chosen for the Capture

## Organization: *object*

### PROPERTIES

**id:** *string* **required**

Unique Identifier of the Organization

**institutionId:** *string* **required**

ID of the Institution to which the Organization belongs

**name:** *string* **required**

Name of the Organization returned

**departmentCount:** *integer (int64)* **required**

Number of Departments that exist for the current Organization

**courseCount:** *integer (int64)* **required**

Number of Courses that exist for the current Organization

**sectionCount:** *integer (int64)* **required**

Number of Sections that exist for the current Organization

**externalId:** *string*

External system identifier

PhoneNumber: *object*

#### PROPERTIES

**region:** *string* **required**

Region to which the Phone Number is assigned. The Region will be used to look up the International Calling Code

#### EXAMPLE

```
"United Kingdom"
```

**number:** *string* **required**

The 10-digit Phone Number

#### EXAMPLE

```
"999-999-9999"
```

ProxyOutbound: *object*

#### PROPERTIES

**host:** *string*

Host Address of the Outbound Proxy Server

**port:** *integer (int32)*

Port Number of the Outbound Proxy Server

**userName:** *string*

User Account to the Outbound Proxy Server

**password:** *string*

Password for the User Account

Room: *object*

#### PROPERTIES

**id:** *string* **required**

Unique Identifier of the Room

**buildingId:** *string* **required**

ID of the Building to which to the Room belongs

**institutionId:** *string* **required**

ID of the Institution to which the Building belongs

**name:** *string* **required**

Name of the Room returned

**roomConfigurationId:** *string*

ID of the Room Configuration. Not Required

**deviceSoftwareVersion:** *string*

Current Software Version on the Lecture Capture Device

#### EXAMPLE

```
"5.5.558149484"
```

**deviceId:** *string*

ID/MAC address for room's capture device

**deviceType:** *string*, x { "PersonalCapture", "ProHardwareCapture", "ProHardwareCapture2", "ProHardwareCapture3" }

The device type, if applicable

**createdAt:** *string* **required**

DateTime value for Creation Date

#### EXAMPLE

```
"YYYY-MM-DD'T'HH:mm:ss.SSS'Z'"
```

**updatedAt:** *string* **required**

DateTime value for Updated Date

#### EXAMPLE

```
"YYYY-MM-DD'T'HH:mm:ss.SSS'Z'"
```

**externalId:** *string*

External system identifier for the current Room

## Schedule: *object*

### PROPERTIES

**id:** *string* **required**

ID of the Schedule

**startDate:** *string* **required**

Start date for capture in ISO 8601 Date format: YYYY-MM-DD

#### EXAMPLE

```
"2017-02-01"
```

**startTime:** *string* **required**

Start time for the capture (to the minute) in ISO 8601 Time format: HH:MM. The time zone for the schedule's linked room will be used

#### EXAMPLE

```
"09:00"
```

**endDate:** *string*

The last date for the capture, in ISO 8601 Date format: YYYY-MM-DD. If this is specified, `daysOfWeek` must also be provided

#### EXAMPLE

```
"2017-05-31"
```

**daysOfWeek:** *string[]*

Days of the week this schedule should capture as 2 letter abbreviations: SU, MO, TU, WE, TH, FR, SA. This is required when `endDate` is specified

#### EXAMPLE

```
"[\\"MO\\", \\"TU\\"]"
```

### ITEMS

*string*

**exclusionDates:** *string[]*

Dates within the scheduled period skip recording in ISO 8601 Date format: YYYY-MM-DD. This is unused when no endDate/daysOfWeek is specified

#### EXAMPLE

```
"[\"2017-04-14\", \"2014-04-17\"]"
```

#### ITEMS

*string*

**durationMinutes:** *integer (int32)* **required**

Duration of each scheduled capture in Minutes

#### EXAMPLE

```
60
```

**sectionId:** *string*

ID of the primary (first) Section in which to publish the Capture

**sections:** *object[]* **required**

List of the Sections in which to Schedule the Capture

#### ITEMS

[ScheduleSection](#)

**name:** *string* **required**

Name of the Schedule

**roomId:** *string* **required**

ID of the Room in which to Schedule the Capture

**instructorId:** *string*

ID of the Instructor

**guestInstructor:** *string*

Name of any Guest Instructor

**shouldCaption:** *boolean* **required**

Boolean value indicating the Capture should be Captioned

**shouldAutoPublish:** *boolean* **required**

Boolean value indicating the Capture should be Automatically Published

**shouldStreamLive:** *boolean* **required**

Boolean value indicating the Capture should be Streamed Live

**input1:** *string*, x { "display", "video", "altvideo" }

Input 1 Logical Graphics Type

**input2:** *string*, x { "display", "video", "altvideo" }

Input 2 Logical Graphics Type

**captureQuality:** *string*, x { "medium", "high", "highest" } **required**

Quality of the Capture

**streamQuality:** *string*, x { "medium", "high", "highest" }

Streaming Quality of the Capture

**externalId:** *string*

External system identifiers

ScheduleAvailability: *object*

#### PROPERTIES

**availability:** *string*, x { "Immediate", "Concrete", "Relative", "Unavailable" } **required**

Availability type

**relativeDelay:** *integer (int64)*

Delay in Days from the start of the capture event for relative availability (Only valid for Relative availability)

**concreteTime:** *string (date)*

Date to make available for concrete availability (Only valid for Concrete availability)

ScheduleSection: *object*

#### PROPERTIES

**sectionId:** *string* **required**

ID of the Section in which to publish the Capture

**availability:** [ScheduleAvailability](#)

Should the Capture be made available to students

## Section: *object*

### PROPERTIES

**id:** *string* **required**

Unique Identifier of the Section

**institutionId:** *string* **required**

ID of the Institution to which the Section belongs

**courseId:** *string* **required**

ID of the Course to which the Section belongs

**termId:** *string* **required**

ID of the Term to which the Section belongs

**scheduleIds:** *string[]* **required**

Array of ID's of Schedules which belong to the current Section

### ITEMS

*string*

**sectionNumber:** *string* **required**

Number to uniquely identify each Section

**instructorId:** *string*

Optional ID of the Instructor by whom the Section will be taught

**description:** *string*

Optional Description of the current Section

**lessonCount:** *integer (int64)* **required**

Deprecated - Number of Lessons that exist for the current Section

### EXAMPLE

15

**instructorCount:** *integer (int64)* **required**

Number of Instructors that exist for the current Section

### EXAMPLE

30

**studentCount:** *integer (int64)* **required**

Number of Students that exist for the current Section

#### EXAMPLE

30

**secondaryInstructorIds:** *string[]*

Array of ID's of the Secondary Instructors for the current Section

#### ITEMS

*string*

**ImscourseIds:** *string[]*

Array of Section Identifiers from any External Learning Management System

#### ITEMS

*string*

**Imscourses:** *string[]*

Array of Section Identifiers from any External Learning Management System

#### ITEMS

*string*

**externalId:** *string*

External system identifier for the current Section

**SoftwareInputs:** *object*

#### PROPERTIES

**audioName:** *string*

Name of the Audio Input

**audioIndex:** *integer (int32)*

Index of the Audio Input

**audioEnabled:** *boolean*

Enabled status of the Audio Input

**displayName:** *string*

Name of the Display Input

**displayIndex:** *integer (int32)*

Index of the Display Input

**displayEnabled:** *boolean*

Enabled status of the Display Input

**videoName:** *string*

Name of the Video Input

**videoIndex:** *integer (int32)*

Index of the Video Input

**videoEnabled:** *boolean*

Enabled status of the Video Input

**altVideoName:** *string*

Name of the Alternative Video Input Configuration

**altVideoIndex:** *integer (int32)*

Index of the Alternative Video Input

**altVideoEnabled:** *boolean*

Enabled status of the Alternative Video Input

Term: *object*

## PROPERTIES

**id:** *string* **required**

Unique Identifier of the Term

**institutionId:** *string* **required**

ID of the Institution to which the Term belongs

**name:** *string* **required**

Name of the Term returned

**session:** *DateRange* **required**

Object consisting of the Start Date and End Date of the Term

**exceptions:** *string[]* **required**

Any Dates within the current Term on which Class will not be held

## ITEMS

*string*

**sectionCount:** *integer (int64)* **required**

Number of Sections that exist for the current Term

**externalId:** *string*

External system identifier for the current Term

Token: *object*

## PROPERTIES

**sharedSecret:** *string*

**consumerKey:** *string*

UpdateExternalId: *object*

## PROPERTIES

**externalId:** *string*

List of external Id for this object, can be empty

UpdateScheduleDeviceOptions: *object*

## PROPERTIES

**roomId:** *string* **required**

ID of the Room in which to Schedule the Capture

**input1:** *string*, x { "display", "video", "altvideo" }

Input for this schedule

**input2:** *string*, x { "display", "video", "altvideo" }

Optional secondary input for this schedule

**captureQuality:** *string*, x { "medium", "high", "highest" } **required**

The capture quality for this schedule

**streamQuality:** *string*, x { "medium", "high", "highest" }

The (optional) quality for the live stream, if it differs from captureQuality

## UpdateScheduleFlags: *object*

### PROPERTIES

**shouldCaption:** *boolean* **required**

Boolean value indicating the Capture should be Captioned

**shouldStreamLive:** *boolean* **required**

Boolean value indicating the Capture should be Streamed Live

## UpdateScheduleInstructors: *object*

### PROPERTIES

**instructorId:** *string*

ID or Email of the Instructor

**guestInstructor:** *string*

Name of any Guest Instructor

## UpdateScheduleName: *object*

### PROPERTIES

**name:** *string*

Name of the Schedule

## UpdateScheduleSection: *object*

### PROPERTIES

**sectionId:** *string*

ID of the Section with which to update the Schedule

**sections:** *object[]*

List of the Sections in which to Schedule the Capture

#### ITEMS

[ScheduleSection](#)

**resolvedSections:** *object[]*

#### ITEMS

## UpdateScheduleTimes: *object*

### PROPERTIES

**startDate:** *string* **required**

Start date for capture in ISO 8601 Date format: YYYY-MM-DD

#### EXAMPLE

```
"2017-02-01"
```

**startTime:** *string* **required**

Start time for the capture (to the minute) in ISO 8601 Time format: HH:MM. The time zone for the schedule's linked room will be used

#### EXAMPLE

```
"09:00"
```

**endDate:** *string*

The last date for the capture, in ISO 8601 Date format: YYYY-MM-DD. If this is specified, daysOfWeek must also be provided

#### EXAMPLE

```
"2017-05-31"
```

**daysOfWeek:** *string[]*

Days of the week this schedule should capture as 2 letter abbreviations. This is required when endDate is specified

#### ITEMS

*string*

**exclusionDates:** *string[]*

Dates within the scheduled period skip recording in ISO 8601 Date format: YYYY-MM-DD

#### EXAMPLE

```
"[\"2017-04-14\", \"2014-04-17\"]"
```

## ITEMS

*string*

**durationMinutes:** *integer (int32)* **required**

Duration of each scheduled capture in Minutes

### EXAMPLE

```
60
```

UpdateSection: *object*

## PROPERTIES

**sectionNumber:** *string* **required**

Number to uniquely identify each Section

### EXAMPLE

```
"ICT102_S1_2017"
```

**description:** *string*

Optional Description of the current Section

### EXAMPLE

```
"First year introductory unit to object-orientated and functional programming"
```

**instructorId:** *string*

ID, Email, or External ID of the Instructor by whom the Section will be taught

### EXAMPLE

```
"Foo.Bar@echo360.com"
```

**externalId:** *string*

External system identifier of the Section

### EXAMPLE

```
"ICT102_S1_2017"
```

## UpdateSectionInstructors: *object*

### PROPERTIES

**secondaryInstructorIds:** *string[]* **required**

Array of ID's of the Secondary Instructors for the current Section

### ITEMS

*string*

## UpdateSectionLmsCourses: *object*

### PROPERTIES

**lmsCourseIds:** *string[]*

Set of Section Identifiers from any External Learning Management System

### ITEMS

*string*

**lmsCourses:** *string[]*

Array of Section Identifiers from any External Learning Management System

### ITEMS

*string*

**validations:** [ValidationGroupUpdateSectionLmsCourses](#)

## User: *object*

### PROPERTIES

**id:** *string* **required**

Unique Identifier of the User

**institutionId:** *string* **required**

ID of the Institution to which the User belongs

**email:** *string* **required**

Email address of the User

### EXAMPLE

```
"Foo.Bar@echo360.com"
```

**timeZone:** *string* **required**

The Time Zone in which the User is located

**EXAMPLE**

```
"US/Eastern"
```

**timeZoneOffsetMinutes:** *integer (int32)* **required**

Offset in minutes to add to UTC to get the local time

**EXAMPLE**

```
-120
```

**firstName:** *string*

First Name of the User. May be blank if the User has not yet accepted the invitation Email

**EXAMPLE**

```
"Foo"
```

**lastName:** *string*

Last Name of the User. May be blank if the User has not yet accepted the invitation Email

**EXAMPLE**

```
"Bar"
```

**status:** *string*

Status of the User.

**EXAMPLE**

```
"Invited"
```

**phoneNumber:** *PhoneNumber*

Phone Number of the User. An object containing the Region and Number

## EXAMPLE

```
"{"region": "US", "number": "000-867-5309"}"
```

**profileImageUrl:** *string*

URL of the User's Profile Image

**roles:** *string[]* **required**

Role(s) of the User within the Institution: Admin, Instructor, Student

## EXAMPLE

```
"[\"Admin\", \"Instructor\", \"Student\"]"
```

## ITEMS

*string*

**ssold:** *string*

A string that identifies the user to an external identity provider

## EXAMPLE

```
"fbar+internet2.edu@subdomain.incommon.org"
```

**externalId:** *string*

External system identifier for the current User

## EXAMPLE

```
"Foo.Bar@echo360.com"
```

UserEnrollment: *object*

## PROPERTIES

**userId:** *string* **required**

ID of the User

**sectionId:** *string* **required**

ID of the Section

**role:** *string*, x { "instructor" , "student" } **required**

Role of the User within the Institution

**institutionId:** *string* **required**

ID of the Institution

**courseId:** *string*

ID of the Course

**termId:** *string*

ID of the Term

ValidationGroupDateTimeRange: *object*

#### PROPERTIES

**first:** [ValidatorDateTimeRange](#)

**rest:** *object[]*

#### ITEMS

[ValidatorDateTimeRange](#)

ValidationGroupLessonInfo: *object*

#### PROPERTIES

**first:** [ValidatorLessonInfo](#)

**rest:** *object[]*

#### ITEMS

[ValidatorLessonInfo](#)

ValidationGroupLtiLinkInfo: *object*

#### PROPERTIES

**first:** [ValidatorLtiLinkInfo](#)

**rest:** *object[]*

#### ITEMS

[ValidatorLtiLinkInfo](#)

ValidationGroupUpdateSectionLmsCourses: *object*

**PROPERTIES**

**first:** *ValidatorUpdateSectionLmsCourses*

**rest:** *object[]*

**ITEMS**

*ValidatorUpdateSectionLmsCourses*

ValidatorDateTimeRange: *object*

ValidatorLessonInfo: *object*

ValidatorLtiLinkInfo: *object*

ValidatorUpdateSectionLmsCourses: *object*