

# API Overview

All of the APIs are organized around [REST](#) - if you've interacted with a RESTful API already, many of the concepts will be familiar to you. We use many standard HTTP features, like HTTP verbs, which can be understood by many HTTP clients. JSON will be returned in all responses from the API, including errors. The APIs are designed to have predictable, straightforward URLs and to use HTTP response codes to indicate API errors.

The base url for all of the endpoints is /api/v1 on your domain.

For a given resource, the available endpoints will be:

- GET /resources : Index for a resource
- GET /resources/:id : Show for one resource
- POST /resources : Create a resource
- PATCH /resource/:id : Update a resource

Some resources can be filtered, i.e.:

- GET students?first\_name=john

For feedback / support, email [help@progettalearn.com](mailto:help@progettalearn.com).

# API Access and Token

In order to access the api, a user must have the “access” checkbox under API in the permission checked. Without this no access will be given, a 403 will be returned on all calls.

Any permissions that the user has will also apply to the api. For instance, if the user is able to view students, /students will return a list of students. If you remove that permission, the endpoint will return a 403 error.

The token can be accessed for a user under My Account -> My Profile in the navigation.

## Authentication

We use simple header authentication, with the email and token sent through the headers. See response codes for the http codes returned.

```
x-user-email: user@example.com  
x-user-token: *****
```

# Endpoints

[API Overview](#)

[API Access and Token](#)

[Authentication](#)

[Endpoints](#)

[Tags](#)

[GET: /tags](#)

[GET: /tags/:id](#)

[Students](#)

[GET /students](#)

[Filters:](#)

[GET /students/:id](#)

[School Programs](#)

[GET /school\\_programs](#)

[GET /school\\_programs/:id](#)

[Tuition Agreements](#)

[GET /tuition\\_agreements](#)

[GET /tuition\\_agreements/:id](#)

[Activity Log \(Notes\)](#)

[GET: /students/:id/student\\_binders/:id/activity\\_log/:id](#)

[GET /students/:id/student\\_binders/:id/activity\\_logs](#)

[POST /students/:id/student\\_binders/:id/activity\\_logs](#)

[Users](#)

[GET /users](#)

[GET /users/:id](#)

[Courses](#)

[GET /school\\_periods/:id/courses](#)

[GET /school\\_periods/:id/courses/:id](#)

[Course Registrations](#)

[GET /student\\_binders/:id/course\\_registrations](#)

[GET /student\\_binders/:id/course\\_registrations/:id](#)

[PATCH /student\\_binders/:id/course\\_registrations/:id](#)

### CoursesLinks

[GET /school\\_periods/:id/courses/:id/course\\_links](#)

[GET /school\\_periods/:id/courses/:id/course\\_links/:id](#)

### School Periods

[GET /school\\_periods](#)

[Filters:](#)

[GET /school\\_periods/:id](#)

### Student Binders

[GET /student\\_binders](#)

[GET /student\\_binders/:id](#)

### Attendance Records

[POST /student\\_binders/:id/course\\_registrations/:id/attendance\\_records](#)

[GET /student\\_binders/:id/course\\_registrations/:id/attendance\\_records](#)

### Assignments

[GET school\\_periods/1/courses/1/assignments](#)

[GET school\\_periods/1/courses/1/assignments/:id](#)

### Response Codes

[GET](#)

[POST](#)

[PATCH](#)

[Errors](#)

# Tags

GET: /tags

GET: /tags/:id

```
{
  "tags": [
    {
      "id": "3",
      "name": "Activity-Log-Tag",
      "taggings_count": "0",
      "description": "",
      "tag_type": "activity_log",
      "staff_id": "",
      "data": "{}"
    },
    {
      "id": "1",
      "name": "Student-Tag",
      "taggings_count": "0",
      "description": "",
      "tag_type": "student",
      "staff_id": "",
      "data": "{}"
    },
    {
      "id": "4",
      "name": "Workflow-Tag",
      "taggings_count": "0",
      "description": "",
      "tag_type": "workflow",
      "staff_id": "",
      "data": "{}"
    }
  ]
}
```

The available options for the tag\_type are:

- student
- workflow
- activity\_log
- content\_library

# Students

GET /students

Filters:

- first\_name
- last\_name
- status\_group
  - incoming
  - active
  - inactive
  - complete

GET /students/:id

```
{
  "id": "22",
  "tag_list": "",
  "email": "student22@progettalearn.com",
  "first_name": "Cheryl",
  "last_name": "Turner",
  "phone": "530-555-1212",
  "gender": "female",
  "created_at": "Dec 27, 2019 12:54 pm",
  "updated_at": "Dec 27, 2019 12:54 pm"
}
```

# School Programs

GET /school\_programs

GET /school\_programs/:id

```
{
  "id": "1",
  "name": "Health Care Plans",
  "created_at": "Dec 27, 2019 12:54 pm",
  "updated_at": "Dec 27, 2019 12:54 pm",
  "data": "{}"
}
```

# Tuition Agreements

GET /tuition\_agreements

GET /tuition\_agreements/:id

```
{
  "id": "1",
  "name": "Standard Agreement",
  "tuition_amount_cents": "1000000",
  "tuition_amount_currency": "USD",
  "admissions_fee_amount_cents": "5000",
  "admissions_fee_amount_currency": "USD",
  "hide_finances_from_student": "false",
  "hide_finances_from_student_message": "",
  "data": "{}"
}
```



# Activity Log (Notes)

GET: /students/:id/student\_binders/:id/activity\_log/:id

```
{
  "id": "9",
  "activityable_type": "StudentBinder",
  "activityable_id": "1",
  "initiator_type": "",
  "initiator_id": "",
  "message": "A message for you",
  "activity_type": "change",
  "data": "",
  "record_changes": "{}",
  "created_at": "Dec 23, 2019 11:04 pm",
  "updated_at": "Dec 23, 2019 11:04 pm",
  "tag_list": ""
}
```

GET /students/:id/student\_binders/:id/activity\_logs

POST /students/:id/student\_binders/:id/activity\_logs

Creates a new activity log, the activity type will be “comment” for all notes created via the API.

Message is required, user\_id and created\_at are optional.

Sample Request:

```
{
  "activity_log": {
    "message": "A message for you",
    "user_id": "2",
    "created_at": "2019-01-01 5:00:00",
  }
}
```

Sample Response:

```
{
  "id": "9",
  "activityable_type": "StudentBinder",
  "activityable_id": "1",
  "initiator_type": "2",
  "initiator_id": "John Smith",
  "message": "A message for you",
  "activity_type": "comment",
  "data": "",
  "record_changes": "{}",
  "created_at": "Jan 1, 2019 5:00 am",
  "updated_at": "Dec 23, 2019 11:04 pm",
  "tag_list": ""
}
```

# Users

GET /users

GET /users/:id

```
{
  "id": "1",
  "email": "user1@progettalearn.com",
  "first_name": "John",
  "last_name": "Smith",
  "full_name": "John Smith",
  "middle_name": "",
  "gender": "male",
  "birthdate": "Jan 1, 1980",
  "phone": "530-555-1212",
  "role": "staff",
  "last_sign_in_at": "",
  "created_at": "Dec 27, 2019 12:54 pm",
  "updated_at": "Dec 27, 2019 12:54 pm"
}
```

Available options for roles are:

- Staff
- Student

# Courses

GET /school\_periods/:id/courses

GET /school\_periods/:id/courses/:id

```
{
  "id": "1",
  "name": "Course 1",
  "body": "",
  "course_type": "online",
  "active": "false",
  "start_date": "Dec 27, 2019",
  "stop_date": "Jan 10, 2020",
  "attendance_type": "none",
  "days": "",
  "times": "",
  "location": "",
  "created_at": "Dec 27, 2019 12:54 pm",
  "updated_at": "Dec 27, 2019 12:54 pm",
  "catalog_course_id": "",
  "multiple_school_periods": "false",
  "grade_type": "graded",
  "data": "{}"
}
```

# Course Registrations

GET /student\_binders/:id/course\_registrations

GET /student\_binders/:id/course\_registrations/:id

```
{
  "id": "18",
  "course_id": "4",
  "course": "Course 4",
  "student_binder_id": "1",
  "student_binder": "Jack Berry",
  "status": "Enrolled",
  "grade": "NaN",
  "letter_grade": "",
  "start_date": "Jan 6, 2020",
  "stop_date": "Jan 20, 2020",
  "final_grade": "",
  "final_letter_grade": "",
  "grade_finalized": "false",
  "created_at": "Jan 6, 2020 11:35 am",
  "updated_at": "Jan 6, 2020 11:35 am"
}
```

PATCH /student\_binders/:id/course\_registrations/:id

Sample Request:

- final\_grade is a float
- final\_letter\_grade is lowercase a-f
- grade\_finalized can be true or false

```
{
  "course_registration": {
    "final_grade": "97",
    "final_letter_grade": "a",
    "grade_finalized": true,
  }
}
```

# CoursesLinks

GET /school\_periods/:id/courses/:id/course\_links

GET /school\_periods/:id/courses/:id/course\_links/:id

```
{
  "course_links": [
    {
      "id": "3",
      "previous_course_id": "13",
      "next_course_id": "5",
      "automatic": "",
      "automatic_assignments": "",
      "student_binders_enrolled": "{}",
      "students_binders_enrolled_at": "",
      "created_at": "Jan  6, 2020 11:35 am",
      "updated_at": "Jan  6, 2020 11:35 am",
      "data": "{}",
      "tag_list": ""
    },
    {
      "id": "6",
      "previous_course_id": "9",
      "next_course_id": "5",
      "automatic": "",
      "automatic_assignments": "",
      "student_binders_enrolled": "{}",
      "students_binders_enrolled_at": "",
      "created_at": "Jan  6, 2020 11:35 am",
      "updated_at": "Jan  6, 2020 11:35 am",
      "data": "{}",
      "tag_list": ""
    }
  ]
}
```



# School Periods

GET /school\_periods

Filters:

- /school\_periods?future=1
- /school\_periods?past=1
- /school\_periods?school\_program\_id=id

GET /school\_periods/:id

```
{
  "school_periods": [
    {
      "id": "2",
      "display_name": "Edgeify",
      "period_type": "2",
      "starts_on": "Oct 7, 2020",
      "stops_on": "May 31, 2021",
      "graduation_date": "May 31, 2021",
      "applications_open_on": "Jan 21, 2020",
      "applications_close_on": "May 31, 2021",
      "international_applications_open_on": "Jan 21, 2020",
      "international_applications_close_on": "May 31, 2021",
      "interviews_open_on": "Jan 21, 2020",
      "interviews_close_on": "May 31, 2021",
      "interview_instructions": "",
      "homework_due_time": "1:00 pm",
      "absence_limit": "50",
      "tardy_after": "10",
      "absent_after": "30",
      "swipe_limit_before": "30",
      "weeks_for_revenue": "33",
      "no_refunds_after_days": "",
      "tuition_fund_id": "1",
      "admissions_fund_id": "1",
      "tuition_amount_cents": "500000",
      "tuition_amount_currency": "USD",
      "admissions_fee_amount_cents": "10000",
      "admissions_fee_amount_currency": "USD",
      "created_at": "Dec 23, 2019 04:05 pm",
    }
  ]
}
```



```
    "updated_at": "Dec 23, 2019 04:05 pm",  
    "homework_status_group_id": "2",  
    "course_registration_status_group_id": "3",  
    "student_status_group_id": "1",  
    "school_program_id": "1",  
    "data": "{}"  
  }  
}
```

# Student Binders

GET /student\_binders

Filters:

- /student\_binders?tuition\_agreement\_id=:id
- /student\_binders?program\_id=:id \*\*500 Internal Service Error
- /student\_binders?school\_period\_id=:id
- /student\_binders?status\_group= [:incoming, :active, :complete, :inactive]

GET /student\_binders/:id

```
{
  "id": "2",
  "first_name": "Paula",
  "last_name": "Cole",
  "email": "student2@progettalearn.com",
  "student_id": "2",
  "school_period_id": "3",
  "tuition_agreement_id": "",
  "status": "Applicant",
  "created_at": "Dec 27, 2019 12:54 pm",
  "updated_at": "Dec 27, 2019 12:54 pm"
}
```

# Attendance Records

Attendance records belong to course registrations, which belong to student binders and can be accessed by the following:

POST

/student\_binders/:id/course\_registrations/:id/attendance\_records

```
{
  "attendance_record": {
    "status": "attended",
    "recorded_at": "2019-12-01",
    "minutes": "34"
  }
}
```

GET

/student\_binders/:id/course\_registrations/:id/attendance\_records

# Assignments

GET school\_periods/1/courses/1/assignments

GET school\_periods/1/courses/1/assignments/:id

Sample Response

```
{
  "id": "1",
  "name": "Assignment One",
  "body": "<div class='trix-content'>Body Content</div> ",
  "hand_out_at": "",
  "due_at": "Jan 3, 2020 12:54 pm",
  "assignment_type": "online",
  "points_available": "10.0",
  "assignable_type": "Course",
  "assignable_id": "3",
  "assignable": "Course 3",
  "displayable_type": "",
  "displayable_id": "",
  "displayable": "",
  "grade_type": "graded",
  "created_at": "Dec 27, 2019 12:54 pm",
  "updated_at": "Dec 27, 2019 12:54 pm"
}
```

# Response Codes

## GET

- Success: 200
- Not Found: 404
- Error: 500

## POST

- Success: 201
- Failure: 400

## PATCH

- Success: 202
- Failure: 400

## Errors

- Not Found: 404
- Not Authorized: 403