

# Final Project: API Spec

## Peer Tutoring API

CS 493: Cloud Application Development

Spring 2022

Oregon State University

Last Update: June 4. 11:50 pm Central

Author: Joel Swenddal

**Deployment URL:** <https://final-peertutor-1215pm.uc.r.appspot.com>

**Account Creation/Login URL:** <https://final-peertutor-1215pm.uc.r.appspot.com> (\* uses Google OAuth)

## CONTENTS

Change log	2
Project overview/Note for graders	2
Data Model	3
Input Value Requirements	4
Create an Appointment	5
Update an Appointment (Full Edit)	8
Edit an Appointment (Partial Edit)	12
Delete an Appointment	16
View an Appointment	18
View Appointments (Collection)	20
Create a Student	23
Update a Student (Full Edit)	26
Edit a Student (Partial Edit)	29
View Students (Collection)	32
View a Student	34
Assign a Student to an Appointment	36
Remove a Student from an Appointment	38
Delete a Student	39
View Users	41
View a User's Appointments	43
Update a User	45

## Change log

Version	Change	Date
1.0	Initial version.	June 4, 2022

## Project Overview

This project provides a Web API for a basic peer-to-peer tutoring service. This service allows Users to sign-up/login through a Google account, and then to use a JWT to authenticate when accessing the API. By signing up, Users can schedule tutoring appointments for themselves, which are then available for other students to book. Students, who do not need to sign up/authenticate, can enter basic information about themselves (a Student record) and then book appointments that have been made available by registered Users (Tutors).

## Note for graders

The API is published externally in Testing form only. Therefore, only test users added directly can access the API with their Google accounts. Per the class instructions, the following users were added (below). However, Google gave a warning regarding the three GTA accounts, saying something was possibly wrong with those accounts. If there are any issues authorizing/authenticating with the API, please contact me!

*cs493gta01@gmail.com*

*cs493gta02@gmail.com*

*cs493gta03@gmail.com*

*gangams@oregonstate.edu*

*kokatea@oregonstate.edu*

*panditb@oregonstate.edu*

## Data Model

The app stores three kinds of entities in Datastore: Users (Tutors), Appointments, and Students. Appointments and Students are non-User entities. Users (who are Tutors) have a one-to-many relationship with Appointments. Appointments have a many-to-many relationship with Students.

### Users

*By signing in through Google and using the provided JWT for authentication, Users are able to set appointments for themselves to tutor students.*

Property	Data Type	Notes
id	Integer	The id of the User (which is regarded as the tutor id). Created on initial sign-in through Google.
firstName	String	First name of User (defaults to null on initial account creation).
lastName	String	Last name of User (defaults to null on initial account creation).
email	String	Email contact of the User (defaults to null on initial account creation).

### Appointments

*A signed-in/authenticated User (a Tutor) can create, update, edit, and delete Appointments for tutoring other students. Appointments are initially set up WITHOUT an associated Student(s). Student(s) are added or removed from the Appointment only through distinct add/remove endpoints.*

Property	Data Type	Notes
id	Integer	The id of the Appointment
subject	String	The course/subject of the tutoring Appointment (can be set to null if it is an open/available appointment).
date	DateTime	The date of the Appointment.
startTime	DateTime	The start time of the Appointment.
endTime	DateTime	The end time of the Appointment.
tutor	Integer	The id of the tutor (a User).
notes	String	Notes/details related to the Appointment.
students	Array of JSON objects	Includes the id of each Student associated with the Appointment (multiple students are allowed).

### Students

*Students do NOT need to register with Google to set up a record, but they cannot create/delete/update/edit Appointments. They can however use the endpoints to assign themselves to an appointment and they can create/delete/update/edit their own student record.*

Property	Data Type	Notes
id	Integer	The id of the Student.
firstName	String	First name of Student.
lastName	String	Last name of Student.
email	String	Email contact of the Student (must be a unique email in the database).
appointments	Array of JSON objects	Includes the id of each Appointment associated with the Student (multiple Appointments are allowed).

## Input Value Requirements

When making POST, PUT and PATCH requests, consider the following input value requirements.

### Users

Property	Data Type	Notes
id	Integer	This is not to be included in the request body. Automatically assigned and cannot be edited.
firstName	String	Defaults to null on initial account creation, but can be added through a PUT.
lastName	String	Defaults to null on initial account creation, but can be added through a PUT.
email	String	Defaults to null on initial account creation, but can be added through a PUT.

### Appointments

Property	Data Type	Notes
id	Integer	This is not to be included in the request body. Automatically assigned and cannot be edited.
subject	String	Can be set to any String value or set to Null (as in the case of just making appointments available).
date	String	This is a DateTime object in the database, but for users to manipulate it, use the string format "YYYY-MM-DD". All other forms are rejected.
startTime	String	This is a DateTime object in the database, but for users to manipulate it, use the string format "HH:MM" with 24-hour time. All other forms are rejected. If startTime is later than endTime, the request will be rejected.
endTime	DateTime	This is a DateTime object in the database, but for users to manipulate it, use the string format "HH:MM" with 24-hour time.. All other forms are rejected. If startTime is later than endTime, the request will be rejected.
tutor	Integer	This is not to be included in the request body.
notes	String	Can be set to any String value or set to Null (as in the case of just making appointments available).
students	Array of JSON objects	This is not to be included in the request body. These associations are only made with the distinct endpoints for adding and removing students from appointments.

### Students

Property	Data Type	Notes
id	Integer	This is not to be included in the request body. Automatically assigned and cannot be edited.
firstName	String	Can be set to any string value.
lastName	String	Can be set to any string value.
email	String	Can be set to any string value that is in the email format "____@____.____" (must be a unique email in the database).
appointments	Array of JSON objects	This is not to be included in the request body. These associations are only made with the distinct endpoints for adding and removing students from appointments.

## Create an Appointment

Allows a JWT authenticated User to create an appointment for themselves acting as a Tutor.

Appointments can be created with Subject and Notes as null if these are not known. Students are NOT to be initially assigned to appointments on creation.

POST /appointments
--------------------

### Request

#### Path Parameters

None

#### Headers

Set the Content-Type header to "application/json"

Set the Accept header to "application/json"

Set the Authorization header to "Bearer <Valid Jwt>"

#### Request Body

Required - Must contain subject, date, startTime, endTime, and notes

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Description	Required?
subject	The subject of the tutoring appointment. Can be set to null.	Yes
date	The date of the appointment (YYYY-MM-DD).	Yes
startTime	The time the appointment begins (HH:MM - 24 hour time)	Yes
endTime	The time the appointment begins (HH:MM - 24 hour time)	Yes
notes	Any details related to the appointment. Can be set to null.	Yes

#### Request Body Example

```
{
  "subject": null,
  "date": "2022-02-08",
  "startTime": "11:00",
  "endTime": "12:00",
  "notes": null
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
---------	-------------	-------

Success	201 Created	
Failure	400 Bad Request	<p>400 status code returned if:</p> <ul style="list-style-type: none"> <li>Request is missing any of the 5 required attributes</li> <li>Invalid attribute names are used (e.g. 'id')</li> <li>Invalid input values are used (e.g. 'name' value is a number)</li> <li>Date format (YYYY-MM-DD) or Time format (HH:MM) is incorrect (24-hour time).</li> <li>startTime is later than endTime.</li> </ul> <p>In these cases, the record will not be created.</p>
Failure	401 Unauthorized	Invalid JWT is being used. Authorization header should be set to "Bearer <Valid Jwt>"
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"
Failure	415 Unsupported Media Type	415 status code is returned if the Content-Type of the body is not <b>application/json</b> .

## Response Examples

\* Datastore will automatically generate an ID and stores it with the entity being created.

### Success

Status: 201 Created

```
{
  "subject": null,
  "date": "2022-02-08T00:00:00.000Z",
  "startTime": "11:00:00",
  "endTime": "12:00:00",
  "notes": null,
  "tutor": "115091805963212517715",
  "students": [],
  "id": "5082266527072256",
  "self":
    "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5082266527072256"
}
```

### Failure

Status: 400 Bad Request (Invalid time input)

```
{
  "Error": "Input values for the time variables have some invalid characteristics
(e.g. start time later than end time)"
}
```

### Failure

Status: 400 Bad Request (Missing required attribute)

```
{  
  "Error": "The request object is missing at least one of the required attributes"  
}
```

#### *Failure*

Status: 400 Bad Request (Invalid input value)

```
{  
  "Error": "Input values for one of the appointment attributes is invalid"  
}
```

#### *Failure*

Status: 401 Unauthorized

```
{  
  "Error": "Authentication Error - A valid JWT must be in the Authorization header.  
Check that your JWT has not expired."  
}
```

#### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Incompatible response type request. Set Accept header to  
application/json"  
}
```

#### *Failure*

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server only accepts Content-Type : application/json; Request body must  
be formatted as JSON"  
}
```

## Update an Appointment (Full Edit)

Allows you to fully replace a record's data, except for the ID value and any content in the Students field (which represents the relationship with the Student entity). Must be authenticated as the User (Tutor) associated with the record (valid Jwt).

PUT /appointments/appointment\_id

### Request

#### Path Parameters

appointment\_id

#### Headers

Set the Content-Type header to "application/json"

Set the Accept header to "application/json"

Set the Authorization header to "Bearer <Valid Jwt>"

#### Request Body

Required - Must contain subject, date, startTime, endTime, and notes

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Description	Required?
subject	The subject of the tutoring appointment. Can be set to null.	Yes
date	The date of the appointment (YYYY-MM-DD).	Yes
startTime	The time the appointment begins (HH:MM - 24 hour time)	Yes
endTime	The time the appointment begins (HH:MM - 24 hour time)	Yes
notes	Any details related to the appointment. Can be set to null.	Yes

#### Request Body Example

```
{
  "subject": "CS 101",
  "date": "2022-11-14",
  "startTime": "12:00",
  "endTime": "13:00",
  "notes": "Student has questions about linked lists"
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
---------	-------------	-------



Success	201 Created	
Failure	400 Bad Request	<p>400 status code returned if:</p> <ul style="list-style-type: none"> <li>Request is missing any of the 5 required attributes</li> <li>Invalid attribute names are used (e.g. 'id')</li> <li>Invalid input values are used (e.g. 'name' value is a number)</li> <li>Date format (YYYY-MM-DD) or Time format (HH:MM) is incorrect (24-hour time).</li> <li>startTime is later than endTime.</li> <li>attempting to edit the Students property (which represents the relationship with the Students entity)</li> </ul> <p>In these cases, the record will not be updated.</p>
Failure	401 Unauthorized	Invalid JWT is being used. Authorization header should be set to "Bearer <Valid Jwt>"
Failure	404 Not found	Invalid appointment ID
Failure	405 Method not allowed	The url must include the ID of the appointment that is being updated.
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"
Failure	415 Unsupported Media Type	415 status code is returned if the Content-Type of the body is not <b>application/json</b> .

## Response Examples

### Success

Status: 201 Created (Updated)

```
{
  "subject": "CS 101",
  "date": "2022-11-14T00:00:00.000Z",
  "startTime": "12:00:00",
  "endTime": "13:00:00",
  "notes": "Student has questions about linked lists",
  "tutor": "115091805963212517715",
  "students": [],
  "id": 5082266527072256,
  "self":
  "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5082266527072256"
}
```

### Failure

Status: 400 Bad Request (Invalid time input)

```
{
  "Error": "Input values for the time variables have some invalid characteristics
(e.g. start time later than end time)"
}
```

```
}
```

#### Failure

Status: 400 Bad Request (Missing required attribute)

```
{  
  "Error": "The request object is missing at least one of the required attributes"  
}
```

#### Failure

Status: 400 Bad Request (Invalid input value)

```
{  
  "Error": "Input values for one of the appointment attributes is invalid"  
}
```

#### Failure

Status: 400 Bad Request (Attempted to edit relationship (Students))

```
{  
  "Error": "Do not include the Students attribute in the body. Use the add/remove  
students from appointment endpoints."  
}
```

#### Failure

Status: 401 Unauthorized

```
{  
  "Error": "Authentication Error - A valid JWT must be in the Authorization header.  
Check that your JWT has not expired."  
}
```

#### Failure

Status: 404 Not found

```
{  
  "Error": "No appointment with this id exists"  
}
```

#### Failure

Status: 405 Method not allowed (Missing appointment id in URL)

```
{  
  "Error": "This operation is not supported on a list of appointment records. Use  
an appointment_id"  
}
```

### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Incompatible response type request. Set Accept header to  
application/json"  
}
```

### *Failure*

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server only accepts Content-Type : application/json; Request body must  
be formatted as JSON"  
}
```

## Edit an Appointment (Partial Edit)

Allows you to edit a subset of an appointment record's data, except for the ID value and any content in the Students field (which represents the relationship with the Student entity). Must be authenticated as the User (Tutor) associated with the record (valid Jwt).

PATCH /appointments/appointment_id
------------------------------------

### Request

#### Path Parameters

appointment\_id

#### Headers

Set the Content-Type header to "application/json"

Set the Accept header to "application/json"

Set the Authorization header to "Bearer <Valid Jwt>"

### Request Body

Required - Must contain subject, date, startTime, endTime, or notes. At least one of these attributes must be included at a minimum.

### Request Body Format

JSON

#### Request JSON Attributes (at least one of these is required)

Name	Description	Required?
subject	The subject of the tutoring appointment. Can be set to null.	No
date	The date of the appointment (YYYY-MM-DD).	No
startTime	The time the appointment begins (HH:MM - 24 hour time). If editing only this, remember that it will be evaluated against the existing endTime, and will be rejected if they are incompatible (e.g. startTime is after endTime)	No
endTime	The time the appointment begins (HH:MM - 24 hour time). If editing only this, remember that it will be evaluated against the existing startTime, and will be rejected if they are incompatible (e.g. endTime is before startTime)	No
notes	Any details related to the appointment. Can be set to null.	No

### Request Body Example

```
{
  "date": "2022-01-01",
  "startTime": "12:00",
  "endTime": "13:00",
  "notes": "Student has questions about homework"
}
```

## Response

### Response Body Format

JSON

### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	400 status code returned if: <ul style="list-style-type: none"><li>• Request is missing any of the 5 required attributes</li><li>• Invalid attribute names are used (e.g. 'id')</li><li>• Invalid input values are used (e.g. 'name' value is a number)</li><li>• Date format (YYYY-MM-DD) or Time format (HH:MM) is incorrect (24-hour time).</li><li>• startTime is later than endTime.</li><li>• attempting to edit the Students property (which represents the relationship with the Students entity)</li></ul> In these cases, the record will not be created.
Failure	401 Unauthorized	Invalid JWT is being used. Authorization header should be set to "Bearer <Valid Jwt>"
Failure	404 Not found	Invalid appointment ID
Failure	405 Method not allowed	The url must include the ID of the appointment that is being updated.
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"
Failure	415 Unsupported Media Type	415 status code is returned if the Content-Type of the body is not <b>application/json</b> .

### Response Examples

#### Success

```
Status: 200 OK
{
  "date": "2022-01-01T00:00:00.000Z",
  "startTime": "12:00:00",
  "endTime": "13:00:00",
  "notes": "Student has questions about homework",
  "tutor": "115091805963212517715",
  "students": [],
  "subject": "Math 231",
  "id": 5752652897976320,
  "self":
  "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5752652897976320"
}
```

#### Failure

Status: 400 Bad Request (Invalid time input)

```
{  
  "Error": "Input values for the time variables have some invalid characteristics  
(e.g. start time later than end time)"  
}
```

#### Failure

Status: 400 Bad Request (Missing required attribute)

```
{  
  "Error": "The request object is missing at least one of the required attributes"  
}
```

#### Failure

Status: 400 Bad Request (Invalid input value)

```
{  
  "Error": "Input values for one of the appointment attributes is invalid"  
}
```

#### Failure

Status: 400 Bad Request (Attempted to edit relationship (Students))

```
{  
  "Error": "Do not include the Students attribute in the body. Use the add/remove  
students from appointment endpoints."  
}
```

#### Failure

Status: 401 Unauthorized

```
{  
  "Error": "Authentication Error - A valid JWT must be in the Authorization header.  
Check that your JWT has not expired."  
}
```

#### Failure

Status: 404 Not found

```
{  
  "Error": "No appointment with this id exists"  
}
```

### *Failure*

Status: 405 Method not allowed (Missing appointment id in URL)

```
{  
  "Error": "This operation is not supported on a list of appointment records. Use  
an appointment_id"  
}
```

### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Incompatible response type request. Set Accept header to  
application/json"  
}
```

### *Failure*

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server only accepts Content-Type : application/json; Request body must  
be formatted as JSON"  
}
```

## Delete an Appointment

Allows you to delete an appointment record. Note that all associations with student records will also be deleted. Must be authenticated as the User (Tutor) associated with the record (valid Jwt).

```
DELETE /appointments/appointment_id
```

### Request

#### Path Parameters

Appointment\_id

#### Headers

Set the Authorization header to "Bearer <Valid Jwt>"

#### Request Body

None

#### Request Body Format

None

### Response

#### Response Body Format

None

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No content	Appointment record successfully deleted. Note that all associations with student records will also be deleted.
Failure	401 Unauthorized	Invalid JWT is being used. Authorization header should be set to "Bearer <Valid Jwt>"
Failure	404 Not found	404 status code is returned if the appt_id value is not found
Failure	405 Method Not Allowed	405 status code is returned if the appt_id value is left out of the url (deleting the whole list of appointments is not allowed).

### Response Examples

#### Success

```
Status: 204 No content
```

#### Failure

```
Status: 401 Unauthorized
{
  "Error": "Authentication Error - A valid JWT must be in the Authorization header.
Check that your JWT has not expired."
}
```



### *Failure*

Status: 404 Not found (Invalid ID parameter)

```
{  
  "Error": "No appointment with this id exists"  
}
```

### *Failure*

Status: 405 Method Not Allowed (when appt\_id is not included in url)

```
{  
  "Error": "This operation is not supported on a list of appointments. Use a  
appt_id"  
}
```

## View an Appointment

Allows you to view an existing appointment if you are authenticated with an appropriate Jwt (for the associated tutor/user). Must be authenticated as the User (Tutor) associated with the record (valid Jwt).

GET /appointments/:appointment_id
-----------------------------------

### Request

#### Path Parameters

Name	Description
appointment_id	ID of the appointment

### Headers

Set the Accept header to "application/json"

Set the Authorization header to "Bearer <Valid Jwt>"

### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid JWT is being used. Authorization header should be set to "Bearer <Valid Jwt>"
Failure	404 Not Found	No appointment with this appointment_id exists
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"

### Response Examples

*Success - (with Accept header set to "application/json")*

Status: 200 OK

```
{
  "id": "5757932864012288",
  "tutor": "115091805963212517715",
  "notes": null,
  "subject": null,
  "startTime": "11:00:00",
  "endTime": "12:00:00",
  "date": "2022-02-08T00:00:00.000Z",
  "students": [],
}
```

```
  "self":  
  "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5757932864012288"  
}
```

#### *Failure*

Status: 401 Unauthorized

```
{  
  "Error": "Authentication Error - A valid JWT must be in the Authorization header.  
Check that your JWT has not expired."  
}
```

#### *Failure*

Status: 404 Not found (Invalid ID parameter)

```
{  
  "Error": "No appointment with this id exists"  
}
```

#### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Incompatible response type request. Set Accept header to  
application/json"  
}
```

## View Appointments (collection)

Allows you to view all appointments which are associated with an authenticated User/Tutor (must supply a valid Jwt in the request). Only those appointments for the respective User/Tutor will be listed.

GET /appointments

### Request

#### Path Parameters

None

#### Headers

Set the Accept header to "application/json"

Set the Authorization header to "Bearer <Valid Jwt>"

#### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid JWT is being used. Authorization header should be set to "Bearer <Valid Jwt>"
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"

#### Response Examples

*Success - (with Accept header set to "application/json")*

Status: 200 OK

```
[
  {
    "appointments": [
      {
        "notes": null,
        "subject": "CS 291",
        "endTime": "13:00:00",
        "tutor": "115091805963212517715",
        "date": "2022-03-21T00:00:00.000Z",
        "startTime": "12:00:00",
        "students": [],
        "id": "5638429693968384",
        "self":
        "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5638429693968384"
      }
    ]
  }
],
```

```
{
  "date": "2022-03-21T00:00:00.000Z",
  "notes": null,
  "startTime": "12:00:00",
  "tutor": "115091805963212517715",
  "students": [],
  "endTime": "13:00:00",
  "subject": "CS 272",
  "id": "5661242479869952",
  "self":
    "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5661242479869952"
},
{
  "date": "2022-03-21T00:00:00.000Z",
  "students": [],
  "subject": "CS 493",
  "notes": null,
  "tutor": "115091805963212517715",
  "endTime": "13:00:00",
  "startTime": "12:00:00",
  "id": "5676594102272000",
  "self":
    "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5676594102272000"
},
{
  "endTime": "15:00:00",
  "startTime": "14:00:00",
  "notes": null,
  "tutor": "115091805963212517715",
  "date": "2022-10-03T00:00:00.000Z",
  "subject": "Math 231",
  "students": [],
  "id": "5697673499770880",
  "self":
    "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5697673499770880"
},
{
  "notes": null,
  "startTime": "12:00:00",
  "tutor": "115091805963212517715",
  "date": "2022-03-21T00:00:00.000Z",
  "students": [],
  "endTime": "13:00:00",
  "subject": "CS 344",
  "id": "5734484288733184",
```

```
      "self":
        "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5734484288733184"
      }
    ],
    "total_appointments": 6,
    "next": "https://final-peertutor-1215pm.uc.r.appspot.com/appointments?page=2&limit=5"
  }
]
```

### *Failure*

```
Status: 401 Unauthorized
{
  "Error": "Authentication Error - A valid JWT must be in the Authorization header.
Check that your JWT has not expired."
}
```

### *Failure*

```
Status: 406 Not Acceptable
{
  "Error": "Incompatible response type request. Set Accept header to
application/json"
}
```

## Create a Student

Creates a new Student record

POST /students

### Request

#### Path Parameters

None

#### Headers

Set the Content-Type header to "application/json"

Set the Accept header to "application/json"

#### Request Body

Required - Must contain firstName, lastName, and email

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Description	Required?
firstName	String representing the first name of the student.	Yes
lastName	String representing the last name of the student.	Yes
email	Email-formatted string representing the contact email of the student. Must be in format "____@__.____". Must also be unique in the database.	Yes

#### Request Body Example

```
{
  "firstName": "Darth",
  "lastName": "Vader",
  "email": "darth@starwars.org"
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	400 status code returned if: <ul style="list-style-type: none"><li>Request is missing any of the 3 required attributes</li><li>Invalid attribute names are used (e.g. 'id')</li><li>Invalid input values are used (e.g. 'firstName' value is a number or email format is invalid)</li></ul>

		In these cases, the record will not be created.
Failure	403 Forbidden	If the same email address is already held in the database for another student, the new one will not be accepted.
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"
Failure	415 Unsupported Media Type	415 status code is returned if the Content-Type of the body is not <b>application/json</b> .

### Response Examples

- \* Datastore will automatically generate an ID and store it with the entity being created.

#### Success

```
Status: 201 Created
{
  "subject": null,
  "date": "2022-02-08T00:00:00.000Z",
  "startTime": "11:00:00",
  "endTime": "12:00:00",
  "notes": null,
  "tutor": "115091805963212517715",
  "students": [],
  "id": "5082266527072256",
  "self":
    "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5082266527072256"
}
```

#### Failure

```
Status: 400 Bad Request (Missing required attribute)

{
  "Error": "The request object is missing at least one of the required attributes"
}
```

#### Failure

```
Status: 400 Bad Request (Invalid input value)

{
  "Error": "Input values for one of the appointment attributes is invalid"
}
```

#### Failure

```
Status: 401 Unauthorized

{
```



```
  "Error": "Authentication Error - A valid JWT must be in the Authorization header.  
Check that your JWT has not expired."  
}
```

#### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Incompatible response type request. Set Accept header to  
application/json"  
}
```

#### *Failure*

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server only accepts Content-Type : application/json; Request body must  
be formatted as JSON"  
}
```

## Update a Student (full edit)

Updates/replaces the original data in a Student record.

PUT /students/student\_id

### Request

Path Parameters

None

### Headers

Set the Content-Type header to “application/json”

Set the Accept header to “application/json”

### Request Body

Required - Must contain firstName, lastName, and email

### Request Body Format

JSON

### Request JSON Attributes

Name	Description	Required?
firstName	String representing the first name of the student.	Yes
lastName	String representing the last name of the student.	Yes
email	Email-formatted string representing the contact email of the student. Must be in format “_____@____._____”. Must also be unique in the database.	Yes

### Request Body Example

```
{
  "firstName": "Anakin",
  "lastName": "Skywalker",
  "email": "anakin@starwars.org"
}
```

### Response

Response Body Format

JSON

### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Updated
Failure	400 Bad Request	400 status code returned if: <ul style="list-style-type: none"><li>Request is missing any of the 3 required attributes</li><li>Invalid attribute names are used (e.g. 'id')</li></ul>

		<ul style="list-style-type: none"> <li>Invalid input values are used (e.g. 'firstName' value is a number or email format is invalid)</li> <li>Attempt to edit Appointments property (this is only editable through the unique endpoint)</li> </ul> <p>In these cases, the record will not be created.</p>
Failure	403 Forbidden	If the same email address is already held in the database for another student, the new one will not be accepted.
Failure	404 Not found	Student id is not found in the database
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"
Failure	415 Unsupported Media Type	415 status code is returned if the Content-Type of the body is not <b>application/json</b> .

### Response Examples

\* Datastore will automatically generate an ID and store it with the entity being created.

#### Success

```
Status: 201 Created (Updated)
{
  "firstName": "Anakin",
  "lastName": "Skywalker",
  "email": "anakinSky@starwars.org",
  "appointments": [],
  "id": 5748302876246016,
  "self":
  "https://final-peertutor-1215pm.uc.r.appspot.com/students/5748302876246016"
}
```

#### Failure

```
Status: 400 Bad Request (Missing required attribute)

{
  "Error": "The request object is missing at least one of the required attributes"
}
```

#### Failure

```
Status: 400 Bad Request (Invalid input value)
{
  "Error": "Input values for one of the student attributes is invalid"
}
```

#### Failure

```
Status: 400 Bad Request (Invalid attribute)
{
```

```
  "Error": "The request object has at least one attribute that is not allowed in a  
student record"  
}
```

#### *Failure*

Status: 400 Bad Request (Attempted to edit relationship (Appointments))

```
{  
  "Error": "Do not include the Appointments attribute in the body. Use the  
add/remove students from appointment endpoints."  
}
```

#### *Failure*

Status: 403 Forbidden

```
{  
  "Error": "There is already another student with this email address. Please use a  
different address"  
}
```

#### *Failure*

Status: 404 Not found

```
{  
  "Error": "No student with this id exists"  
}
```

#### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Incompatible response type request. Set Accept header to  
application/json"  
}
```

#### *Failure*

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server only accepts Content-Type : application/json; Request body must  
be formatted as JSON"  
}
```

## Edit a Student (partial edit)

Edits a Student record.

PATCH /students/student_id
----------------------------

### Request

Path Parameters

None

### Headers

Set the Content-Type header to "application/json"

Set the Accept header to "application/json"

### Request Body

Required - Must contain firstName, lastName, or email

### Request Body Format

JSON

### Request JSON Attributes

Name	Description	Required?
firstName	String representing the first name of the student.	No
lastName	String representing the last name of the student.	No
email	Email-formatted string representing the contact email of the student. Must be in format "____@__.____". Must also be unique in the database.	No

### Request Body Example

```
{
  "firstName": "Leia",
  "lastName": "Skywalker"
}
```

### Response

Response Body Format

JSON

### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Edited
Failure	400 Bad Request	400 status code returned if: <ul style="list-style-type: none"><li>Request is missing one of the 3 required attributes</li><li>Invalid attribute names are used (e.g. 'id')</li><li>Invalid input values are used (e.g. 'firstName' value is a number or email format is invalid)</li></ul>

		<ul style="list-style-type: none"> <li>Attempt to edit Appointments property (this is only editable through the unique endpoint) In these cases, the record will not be created.</li> </ul>
Failure	403 Forbidden	If the same email address is already held in the database for another student, the new one will not be accepted.
Failure	404 Not found	Student id is not found in the database
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"
Failure	415 Unsupported Media Type	415 status code is returned if the Content-Type of the body is not <b>application/json</b> .

### Response Examples

\* Datastore will automatically generate an ID and store it with the entity being created.

#### Success

Status: 200 OK (Edited)

```
{
  "firstName": "Leia",
  "lastName": "Skywalker",
  "email": "quiGonJin@starwars.org",
  "appointments": [],
  "id": 5726263989764096,
  "self":
    "https://final-peertutor-1215pm.uc.r.appspot.com/students/5726263989764096"
}
```

#### Failure

Status: 400 Bad Request (Missing required attribute)

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

#### Failure

Status: 400 Bad Request (Invalid input value)

```
{
  "Error": "Input values for one of the student attributes is invalid"
}
```

#### Failure

Status: 400 Bad Request (Invalid attribute)

```
{
  "Error": "The request object has at least one attribute that is not allowed in a student record"
}
```

```
}
```

#### *Failure*

Status: 400 Bad Request (Attempted to edit relationship (Appointments))

```
{
  "Error": "Do not include the Appointments attribute in the body. Use the
add/remove students from appointment endpoints."
}
```

#### *Failure*

Status: 403 Forbidden

```
{
  "Error": "There is already another student with this email address. Please use a
different address"
}
```

#### *Failure*

Status: 404 Not found

```
{
  "Error": "No student with this id exists"
}
```

#### *Failure*

Status: 406 Not Acceptable

```
{
  "Error": "Incompatible response type request. Set Accept header to
application/json"
}
```

#### *Failure*

Status: 415 Unsupported Media Type

```
{
  "Error": "Server only accepts Content-Type : application/json; Request body must
be formatted as JSON"
}
```

## View Students (collection)

Allows you to view all students (not protected by authorization/authentication).

GET /students

### Request

#### Path Parameters

None

#### Headers

Set the Accept header to “application/json”

### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	The Accept header needs to be set to “application/json”

### Response Examples

#### Success

Status: 200 OK

```
[
  {
    "students": [
      {
        "firstName": "Admiral",
        "appointments": [],
        "lastName": "Ackbar",
        "email": "rebel4ever@starwars.org",
        "id": "5144110029602816",
        "self":
"https://final-peertutor-1215pm.uc.r.appspot.com/students/5144110029602816"
      },
      {
        "appointments": [],
        "lastName": "Skywalker",
        "firstName": "Anakin",
        "email": "anakinSky@starwars.org",
        "id": "5172503789961216",
```



```

        "self":
"https://final-peertutor-1215pm.uc.r.appspot.com/students/5172503789961216"
    },
    {
        "appointments": [],
        "lastName": "Calrissian",
        "email": "cloudCity@starwars.org",
        "firstName": "Lando",
        "id": "5632046399488000",
        "self":
"https://final-peertutor-1215pm.uc.r.appspot.com/students/5632046399488000"
    },
    {
        "lastName": "Skywalker",
        "email": "quiGonJin@starwars.org",
        "appointments": [],
        "firstName": "Leia",
        "id": "5726263989764096",
        "self":
"https://final-peertutor-1215pm.uc.r.appspot.com/students/5726263989764096"
    },
    {
        "appointments": [],
        "email": "jengo@starwars.org",
        "firstName": "Bobb",
        "lastName": "Fett",
        "id": "5739917086818304",
        "self":
"https://final-peertutor-1215pm.uc.r.appspot.com/students/5739917086818304"
    }
],
"total_students": 6,
"next": "https://final-peertutor-1215pm.uc.r.appspot.com/students?page=2&limit=5"
}
]

```

### Failure

Status: 406 Not Acceptable

```

{
  "Error": "Incompatible response type request. Set Accept header to
application/json"
}

```

## View a Student

Allows you to view a student record (not protected by authorization/authentication)

GET /students/student\_id

### Request

#### Path Parameters

None

#### Headers

Set the Accept header to “application/json”

#### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not found	No student with this id was found in the database
Failure	406 Not Acceptable	The Accept header needs to be set to “application/json”

### Response Examples

#### Success

```
Status: 200 OK
{
  "lastName": "Skywalker",
  "email": "anakinSky@starwars.org",
  "firstName": "Anakin",
  "appointments": [],
  "id": "5172503789961216",
  "self":
    "https://final-peertutor-1215pm.uc.r.appspot.com/students/5172503789961216"
}
```

#### Failure

```
Status: 404 Not found
{
  "Error": "No student with this id exists"
}
```

### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Incompatible response type request. Set Accept header to  
application/json"  
}
```

## Assign a Student to an Appointment

Associate an Appointment record with a Student record

PUT /appointments/appointment\_id/student\_id

### Request

#### Path Parameters

Name	Description
appointment_id	ID of the appointment
student_id	ID of the student

### Request Body

None

### Response

#### Response Body Format

JSON

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Association made between appointment and student
Failure	404 Not Found	Appointment or student id is not found in the database
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"

### Response Examples

#### Success

Status: 201 Created (Created relationship)

```
{
  "id": 5697673499770880,
  "date": "2022-10-03T00:00:00.000Z",
  "startTime": "14:00:00",
  "endTime": "15:00:00",
  "notes": null,
  "tutor": "115091805963212517715",
  "subject": "Math 231",
  "students": [
    {
      "id": "5172503789961216",
      "self":
        "https://final-peertutor-1215pm.uc.r.appspot.com/students/5172503789961216"
    }
  ],
}
```

```
  "self":  
    "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5697673499770880"  
}
```

#### *Failure*

Status: 404 Not found

```
{  
  "Error": "No appointment with this id exists"  
}
```

#### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Incompatible response type request. Set Accept header to  
application/json"  
}
```

## Remove a Student from an Appointment

Deletes the relationship between a student and an appointment

```
DELETE /appointments/appointment_id/student_id
```

### Request

#### Path Parameters

Name	Description
appointment_id	ID of the appointment
student_id	ID of the student

#### Request Body

None

#### Request Body Format

None

### Response

#### Response Body Format

None

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No content	Association between appointment and student removed
Failure	404 Not Found	Appointment or student id is not found in the database
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"

#### Response Examples

##### Success

```
Status: 204 No content (Deleted relationship)
```

##### Failure

```
Status: 404 Not found
{
  "Error": "No appointment with this id exists"
}
```

##### Failure

```
Status: 406 Not Acceptable
{
  "Error": "Incompatible response type request. Set Accept header to
application/json"
}
```

## Delete a Student

Allows you to delete a student record. Note that all associations with appointment records will also be deleted.

```
DELETE /students/student_id
```

### Request

#### Path Parameters

appointment\_id

#### Request Body

None

#### Request Body Format

None

### Response

#### Response Body Format

None

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No content	Student record successfully deleted. Note that all associations with appointment records will also be deleted.
Failure	404 Not found	404 status code is returned if the student_id value is not found
Failure	405 Method Not Allowed	405 status code is returned if the student_id value is left out of the url (deleting the whole list of students is not allowed).

### Response Examples

#### Success

```
Status: 204 No content
```

#### Failure

```
Status: 404 Not found (Invalid ID parameter)
{
  "Error": "No student with this id exists"
}
```

#### Failure

```
Status: 405 Method Not Allowed (when appt_id is not included in url)
{
```

```
"Error": "This operation is not supported on a list of students. Use a  
student_id"  
}
```



## View Users (collection)

Allows you to the list of registered users. Users are created on initial sign-in through Google. The firstName, lastName, and email properties are initialized to null. They can be updated through an edit (see below).

GET /users
------------

### Request

#### Path Parameters

None

### Headers

Set the Accept header to “application/json”

### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	The Accept header needs to be set to “application/json”

### Response Examples

#### Success

```
Status: 200 OK
[
  {
    "id": "115091805963212517715",
    "firstName": null,
    "lastName": null,
    "email": null
  },
  {
    "id": "103910897216606416047",
    "firstName": "Joel",
    "lastName": "Swenddal",
    "email": "testmail@gmail.com"
  }
]
```

### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Incompatible response type request. Set Accept header to  
application/json"  
}
```

## View a User's Appointments

Allows you to view the Appointments that are registered for a User (Tutor). Must be the authenticated User to view.

GET /users/user\_id/appointments

### Request

#### Path Parameters

None

#### Headers

Set the Accept header to "application/json"

Set the Authorization header to "Bearer <Valid Jwt>"

#### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid JWT is being used. Authorization header should be set to "Bearer <Valid Jwt>"
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"

#### Response Examples

##### Success

```
Status: 200 OK
[
  {
    "endTime": "13:00:00",
    "subject": "CS 291",
    "startTime": "12:00:00",
    "date": "2022-03-21T00:00:00.000Z",
    "notes": null,
    "students": [],
    "tutor": "115091805963212517715",
    "id": "5638429693968384",
    "self":
    "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5638429693968384"
  },
]
```

```
{
  "startTime": "12:00:00",
  "notes": null,
  "endTime": "13:00:00",
  "subject": "CS 272",
  "students": [],
  "tutor": "115091805963212517715",
  "date": "2022-03-21T00:00:00.000Z",
  "id": "5661242479869952",
  "self":
    "https://final-peertutor-1215pm.uc.r.appspot.com/appointments/5661242479869952"
}
```

#### *Failure*

Status: 401 Unauthorized

```
{
  "Error": "Authentication Error - A valid JWT must be in the Authorization header.
Check that your JWT has not expired."
}
```

#### *Failure*

Status: 406 Not Acceptable

```
{
  "Error": "Incompatible response type request. Set Accept header to
application/json"
}
```

## Update a User

Edits the original data in a User record (originally set to null). Must be the authenticated User (valid Jwt).

PATCH /users/user_id
----------------------

### Request

Path Parameters

None

### Headers

Set the Content-Type header to "application/json"

Set the Accept header to "application/json"

Set the Authorization header to "Bearer <Valid Jwt>"

### Request Body

Required - Must contain firstName, lastName, or email

### Request Body Format

JSON

### Request JSON Attributes

Name	Description	Required?
firstName	String representing the first name of the student.	No
lastName	String representing the last name of the student.	No
email	Email-formatted string representing the contact email of the student. Must be in format "____@__.____". Must also be unique in the database.	No

### Request Body Example

```
{
  "id": "115091805963212517715",
  "firstName": "Joel",
  "lastName": "Swenddal",
  "email": "testmail@gmail.com"
}
```

### Response

Response Body Format

JSON

### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Updated
Failure	400 Bad Request	400 status code returned if: <ul style="list-style-type: none"><li>Request is missing any of the 3 required attributes</li></ul>

		<ul style="list-style-type: none"> <li>Invalid attribute names are used (e.g. 'id')</li> <li>Invalid input values are used (e.g. 'firstName' value is a number or email format is invalid)</li> <li>Attempt to edit Appointments property (this is only editable through the unique endpoint)</li> </ul> <p>In these cases, the record will not be created.</p>
Failure	401 Unauthorized	Invalid JWT is being used. Authorization header should be set to "Bearer <Valid Jwt>"
Failure	403 Forbidden	You must be authenticated as the User to update the User's record.
Failure	406 Not Acceptable	The Accept header needs to be set to "application/json"
Failure	415 Unsupported Media Type	415 status code is returned if the Content-Type of the body is not <b>application/json</b> .

### Response Examples

\* Datastore will automatically generate an ID and store it with the entity being created.

#### Success

Status: 201 Created (Updated)

```
{
  "id": "115091805963212517715",
  "firstName": "Joel",
  "lastName": "Swenddal",
  "email": "testmail@gmail.com"
}
```

#### Failure

Status: 400 Bad Request (Missing required attribute)

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

#### Failure

Status: 400 Bad Request (Invalid input value)

```
{
  "Error": "Input values for one of the user attributes is invalid"
}
```

#### Failure

Status: 400 Bad Request (Invalid attribute)

```
{
  "Error": "The request object has at least one attribute that is not allowed in a user record"
}
```

```
}
```

#### *Failure*

Status: 401 Unauthorized

```
{  
  "Error": "Authentication Error - A valid JWT must be in the Authorization header.  
Check that your JWT has not expired."  
}
```

#### *Failure*

Status: 403 Forbidden

```
{  
  "Error": "You cannot update a user record if you are not authenticated as that  
user"  
}
```

#### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Incompatible response type request. Set Accept header to  
application/json"  
}
```

#### *Failure*

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server only accepts Content-Type : application/json; Request body must  
be formatted as JSON"  
}
```