Application Data Model

The app stores three entities in Datastore – Teacher, Supply, Student

Teacher

Property	Data Type	Notes
id	Integer	Teacher id - the key that Datastore automatically generates
teacher_autho	String	The teacher_id that is provided by auth0 – the sub attribute (subject whom JWT token refers to)
name	String	Name of user
email	String	Email of user
supplies_needed	Array of objects with key-value pair for supply_id	The array stores the the collection of supplies that the user needs for their classroom.
students	Array of objects with key-value pair for student_id	The array stores the the collection of the user's students.

Supply

Property	Data Type	Notes
id	Integer	Supply id - the key that Datastore automatically generates
teacher_autho	String	The teacher (user) that this supply belongs to – the sub attribute (subject whom JWT token refers to)
description	String	The description of the supply
num_needed	Integer	The number of items of this supply the user needs
num_supplied	Integer	The number of items of this supply that have been donated by students
students	Array of objects with key-value pair for student_id	The array stores the collection of students who donate this supply

Student

Property	Data Type	Notes
id	Integer	Student id - the key that Datastore automatically generates
teacher_autho	String	The teacher (user) that this student has – the sub attribute (subject whom JWT token refers to)
first_name	String	First name of student
last_name	String	Last name of student
class_period	Integer	The class period the student has with the teacher (user)
supply_donated	String	The supply the student donates – a student can donate at most one supply. If the student does not donate an item, the value is set to NULL.

API Specification

Create a Teacher

Allows you to create a new teacher.

```
POST /teachers
```

Request

Path Parameters

None

Request Body Format

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of user	Yes
email	Email of user	Yes
Teacher_autho	Auth0 id of	Yes

Request Body Example

```
{
 "name": "John",
 "email": "john@gmail.com",
 "teacher_autho": "auth0|62996621dc4dea0068d860aa"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. The app sends back this value in the response body as shown in the example.

Success

Status: 201 Created

```
{
    "supplies_needed": [],
    "name": "John",
    "teacher_autho": "auth0|62996621dc4dea0068d860aa",
    "email": "john@gmail.com",
    "students": [],
    "id": "5673966991572992"
}
```

Get a Teacher

Allows you to get an existing teacher

GET /teachers/:teacher_id

Request

Path Parameters

Name	Description
teacher_id	id of the teacher (Integer)

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No teacher with this teacher_id exists

Response Examples

Success

Status: 200 OK

```
{
    "supplies_needed": [],
    "teacher_autho": "auth0|62996621dc4dea0068d860aa",
    "email": "john@gmail.com",
    "name": "John",
    "students": [],
    "id": "5673966991572992",
    "self": "http://localhost:8080/teachers/5673966991572992"
}
```

Failure

Status: 404 Not Found

```
{
"Error":  "No teacher with this teacher_id exists"
}
```

List all Teachers

Lists all the teachers. This endpoint is unprotected.

```
GET /teachers
```

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK

```
[
    {
        "students": [],
        "name": "John",
        "teacher_autho": "auth0|62996621dc4dea0068d860aa",
        "email": "john@gmail.com",
        "supplies_needed": [],
        "id": "5673966991572992"
    },
    {
        "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
        "email": "wallace@cheese.com",
        "supplies_needed": [],
        "students": [],
        "name": "wallace",
        "id": "5740073718906880"
    }
]
```

Put a Teacher

This method is not accepted.

```
PUT /teachers
```

Response Statuses

Outcome	Status Code	Notes
Failure	405 Method not allowed	

Failure

Status: 405 Method Not Allowed

```
{
    "Error": "Route only accepts GET or POST methods"
}
```

Create a Supply

Allows you to create a new supply. This supply will be related to a teacher and the teacher_autho attribute will be assigned from the JWT. JWT must belong to a user who has a teacher account. Otherwise, the supply will not be created.

```
POST /supplies
```

Request

Path Parameters

None

Request Authorization

Required (Bearer Token)

Request Body Format

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
description	Description of supply item	Yes
num_needed	Number of supply items needed by teacher(user)	Yes

Request Body Example

```
{
 "description": "scissors",
 "num_needed": 10
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 2 required attributes or if num_needed is not a non-negative integer, the supply will not be created, and 400 status code will be returned.
Failure	400 Bad Request	If the JWT token belongs to a user that does not have a teacher account, the supply will not be created, and 400 status code will be returned.
Failure	401 Unauthorized	If the request does not include the bearer token authorization, the supply will not be created, and 401 status code will be returned.
Failure	406 Not Acceptable	If the response of type JSON is not acceptable, then this

		406 status code is returned.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. The app sends back this value in the response body as shown in the example.

Success

Status: 201 Created

```
{
    "num_needed": 10,
    "students": [],
    "description": "tissue box",
    "num_supplied": 0,
    "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
    "id": "5751558595674112",
    "self": "http://localhost:8080/supplies/5751558595674112"
}
```

Failure

Status: 400 Bad Request

```
{
    "Error": "The request object is missing at least one of the required attributes or
 the data provided is invalid."
}
```

Failure

Status: 400 Bad Request

```
{
    "Error": "This user does not yet have a teacher account."
}
```

Failure

Status: 401 Unauthorized

```
{
"Error":  "Missing or invalid JWT token."
}
```

Failure

Status: 406 Not Acceptable

```
{
    "Error": "Not Acceptable"
}
```

Get a Supply

Allows you to get an existing supply. The JSON body includes a link to the entity.

```
GET /supplies/:supply_id
```

Request

Path Parameters

Name	Description
supply_id	ID of the supply

Request Authorization

Required (Bearer Token)

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No supply with this supply_id exists
Failure	401 Unauthorized	If the request does not include the bearer token authorization, the supply will not be created, and 401 status code will be returned.
Failure	403 Forbidden	If the request contains a bearer token, but it is not that of the teacher of the supply, then the 403 status code will be returned.

Response Examples

Success

Status: 200 OK

```
{
    "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
    "description": "tissue boxes",
    "num_needed": 19,
    "num_supplied": 0,
    "students": [],
    "id": "5691975521009664",
    "self": "http://localhost:8080/supplies/5691975521009664"
}
```

Failure

Status: 401 Unauthorized

```
{
```

```
"Error":  "Missing or invalid JWT token."
}
```

Failure

```
Status: 403 Forbidden
{
    "Error": "Unauthorized"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No supply with this supply_id exists"
}
```

List all Supplies

List all the supplies that are related to the teacher, the total number of supplies in the collection, and implements pagination. Returns 5 supplies per page and includes a next link to get the next page of results, unless there are no more pages of result. If the teacher does not have any supplies, then an empty array is returned.

Pagination is implemented using cursors.

GET /supplies

Request

Path Parameters

None

Request Body

None

Request Authorization

Required (Bearer Token)

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK

```
{
    "supplies": [
        {
            "students": [],
            "num_supplied": 0,
            "num_needed": 10,
            "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
            "description": "scissors",
            "id": "5691975521009664"
        },
        {
            "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
            "description": "dry erase markers",
            "students": [],
            "num_needed": 25,
            "num_supplied": 0,
            "id": "5726263989764096"
        }
    ],
    "total": 2
}
```

Delete a Supply

Allows the authorized user can delete a supply. If the supply being deleted has a relationship with a student, then the supply will be removed from the student.

DELETE /supplies/:supply_id

Request

Path Parameters

Name	Description
supply_id	ID of the supply

Request Body

None

Request Authorization

Required (Bearer Token)

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	If the request does not include the bearer token authorization, the supply will not be able to be deleted, and 401 status code will be returned.
Failure	403 Forbidden	If the request contains a bearer token, but it is not that of the teacher of the supply, then the 403 status code will be returned.
Failure	404 Not Found	No supply with this supply_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
    "Error": "No supply with this supply_id exists"
}
```

Failure

Status: 403 Forbidden

```
{
    "Error": "Unauthorized"
}
```

Failure

Status: 404 Not Found

```
{
"Error":  "No supply with this supply_id exists"
}
```

Put a Supply

Allows an authorized user to update a supply – the following attributes are all updated: description, num_needed, num_supplied is set to 0, and students is set to an empty array. If any students are related to the supply, the supply_donated field is set to null of these students.

Note that description and num_needed are both required.

PUT /supplies/:supply_id

Request

Path Parameters

Name	Description
supply_id	ID of the supply

Request Authorization

Required (Bearer Token)

Request Body Format

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
description	Description of supply item	Yes
num_needed	Number of supply items needed by teacher(user)	Yes

Request Body Example

```
{
 "description": "tissue box",
 "num_needed": 10
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing any of the 2 required attributes or if num_needed is not a non-negative integer, the supply will not be created, and 400 status code will be returned.
Failure	401 Unauthorized	If the request does not include the bearer token authorization, the supply will not be created, and 401 status code will be returned.
Failure	403 Forbidden	If the request contains a bearer token, but it is not that of

		the teacher of the supply, then the 403 status code will be returned.
Failure	404 Not Found	No supply with this supply_id exists

Response Examples

Success

Status: 201 Created

```
{
    "num_needed": 10,
    "students": [],
    "description": "tissue box",
    "num_supplied": 0,
    "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
    "id": "5751558595674112",
    "self": "http://localhost:8080/supplies/5751558595674112"
}
```

Failure

Status: 400 Bad Request

```
{
    "Error": "The request object is missing at least one of the required attributes or
 the data provided is invalid."
}
```

Failure

Status: 401 Unauthorized

```
{
"Error":  "Missing or invalid JWT token."
}
```

Failure

Status: 403 Forbidden

```
{
    "Error": "Unauthorized"
}
```

Failure

Status: 404 Not Found

```
{
"Error":  "No supply with this supply_id exists"
}
```

Failure

```
Status: 406 Not Acceptable
{
    "Error": "Not Acceptable"
}
```

Patch a Supply

Allows an authorized user to update a supply – it is possible to update the following attributes: description and/or num_needed. However, num_needed is not allowed to be a number less than num_supplied.

The attributes num_supplied and students can be changed using the /supplies/:supply_id/students/:student_id routes.

```
PATCH /supplies
```

Request

Path Parameters

Name	Description
supply_id	ID of the supply

Request Authorization

Required (Bearer Token)

Request Body Format

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
description	Description of supply item	No
num_needed	Number of supply items needed by teacher(user)	No

Request Body Example

```
{
 "description": "tissue box",
 "num_needed": 10
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	200 OK	
Failure	401 Unauthorized	If the request does not include the bearer token authorization, the supply will not be created, and 401 status code will be returned.
Failure	403 Forbidden	If the request contains a bearer token, but it is not that of the teacher of the supply, then the 403 status code will be returned.
Failure	404 Not Found	No supply with this supply_id exists

Response Examples

Success

Status: 201 Created

```
{
    "num_needed": 10,
    "students": [],
    "description": "tissue box",
    "num_supplied": 0,
    "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
    "id": "5751558595674112",
    "self": "http://localhost:8080/supplies/5751558595674112"
}
```

Failure

Status: 401 Unauthorized

```
{
"Error":  "Missing or invalid JWT token."
}
```

Failure

Status: 403 Forbidden

```
{
    "Error": "Unauthorized"
}
```

Failure

Status: 404 Not Found

```
{
"Error":  "No supply with this supply_id exists"
}
```

Create a Student

Allows you to create a new student. This student will be related to a teacher and the teacher_autho attribute will be assigned from the JWT. JWT must belong to a user who has a teacher account. Otherwise, the student will not be created.

```
POST /students
```

Request

Path Parameters

None

Request Authorization

Required (Bearer Token)

Request Body Format

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
first_name	Student first name	Yes
last_name	Student last name	Yes
class_period	Class period of student	Yes

Request Body Example

```
{
  "first_name": "Mia",
  "last_name": "Fisher",
  "class_period": 2
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the student will not be created, and 400 status code will be returned.
Failure	400 Bad Request	If the JWT token belongs to a user that does not have a teacher account, the student will not be created, and 400 status code will be returned.

Failure	401 Unauthorized	If the request does not include the bearer token authorization, the student will not be created, and 401 status code will be returned.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. The app sends back this value in the response body as shown in the example.

Success

Status: 201 Created

```
{
    "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
    "first_name": "Mia",
    "supply_donated": null,
    "class_period": 2,
    "last_name": "Fisher",
    "id": "5700228501995520"
}
```

Failure

Status: 400 Bad Request

```
{
    "Error": "The request object is missing at least one of the required attributes or the data provided is invalid."
}
```

Failure

Status: 400 Bad Request

```
{
    "Error": "This user does not yet have a teacher account."
}
```

Failure

Status: 401 Unauthorized

```
{
"Error":  "Missing or invalid JWT token."
}
```

Failure

Status: 406 Not Acceptable

```
{
    "Error": "Not Acceptable"
}
```

Get a Student

Allows you to get an existing student. The JSON body includes a link to the entity.

```
GET /students/:student_id
```

Request

Path Parameters

Name	Description
student_id	ID of the student

Request Authorization

Required (Bearer Token)

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 student_id exists
Failure	401 Unauthorized	If the request does not include the bearer token authorization, the student will not be created, and 401 status code will be returned.
Failure	403 Forbidden	If the request contains a bearer token, but it is not that of the teacher of the student, then the 403 status code will be returned.

Response Examples

Success

Status: 200 OK

```
{
    "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
    "first_name": "Mia",
    "supply_donated": null,
    "class_period": 2,
    "last_name": "Fisher",
    "id": "5700228501995520",
    "self": "http://localhost:8080/students/5700228501995520"
}
```

Failure

Status: 401 Unauthorized

```
{
```

```
"Error":  "Missing or invalid JWT token."
}
```

Failure

Status: 403 Forbidden

```
{
    "Error": "Unauthorized"
}
```

Failure

Status: 404 Not Found

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

List all Students

List all the students that are related to the teacher, the total number of students in the collection, and implements pagination. Returns 5 students per page and includes a next link to get the next page of results, unless there are no more pages of result. If the teacher does not have any students, then an empty array is returned.

Pagination is implemented using cursors.

GET /students

Request

Path Parameters

None

Request Body

None

Request Authorization

Required (Bearer Token)

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK

```
{
    "students": [
        {
            "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
            "last_name": "Fisher",
            "class_period": 2,
            "first_name": "Mia",
            "supply_donated": null,
            "id": "5653635052601344"
        },
        {
            "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
            "first_name": "Mia",
            "last_name": "Fisher",
            "supply_donated": null,
            "class_period": 2,
            "id": "5669079989878784"
        },
        {
            "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
            "last_name": "Fisher",
            "first_name": "Mia",
            "class_period": 2,
            "supply_donated": null,
            "id": "5694955355897856"
        },
        {
            "class_period": 2,
            "first_name": "Mia",
            "supply_donated": null,
            "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
            "last_name": "Fisher",
            "id": "5700228501995520"
        },
        {
            "class_period": 2,
            "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
            "last_name": "Fisher",
            "first_name": "Mia",
            "supply_donated": null,
            "id": "5715241090416640"
        }
    ],
    "next": "http://localhost:8080/students?cursor=CjYSMGoYc35maW5hbC1wcm9qZWN0LWZpc2h
lYWxpchQLEgdTdHVkZW50GICAgJi0v5MKDBgAIAA=",
    "total": 6
}
```

Delete a Student

Allows the authorized user can delete a student. If the student being deleted has a relationship with a supply, then the student will be removed from the array of students of the supply.

```
DELETE /students/:student_id
```

Request

Path Parameters

Name	Description
student_id	ID of the student

Request Body

None

Request Authorization

Required (Bearer Token)

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	If the request does not include the bearer token authorization, the supply will not be able to be deleted, and 401 status code will be returned.
Failure	403 Forbidden	If the request contains a bearer token, but it is not that of the teacher of the supply, then the 403 status code will be returned.
Failure	404 Not Found	No student with this student_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized
{
"Error":  "Missing or invalid JWT token."
}
```

Failure

```
Status: 403 Forbidden
```

```
{
    "Error": "Unauthorized"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No supply with this supply_id exists"
}
```

Put a Student

Allows an authorized user to update a student – the following attributes are all updated: first_name, last_name, class_period, and supply_donated is set to null. If a supply was related to this student, the student is removed from the supply’s array of students.

Note that first_name, last_name, class_period are required.

PUT /students/:student_id

Request

Path Parameters

Name	Description
student_id	ID of the student

Request Authorization

Required (Bearer Token)

Request Body Format

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
first_name	Student first name	Yes
last_name	Student last name	Yes
class_period	Student class period	Yes

Request Body Example

```
{
    "class_period": 2,
    "first_name": "Alice",
    "last_name": "Fisher"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes the student will not be created, and 400 status code will be returned.
Failure	401 Unauthorized	If the request does not include the bearer token authorization, the student will not be created, and 401

		status code will be returned.
Failure	403 Forbidden	If the request contains a bearer token, but it is not that of the teacher of the student, then the 403 status code will be returned.
Failure	404 Not Found	No student with this student_id exists

Response Examples

Success

Status: 201 Created

```
{
    "first_name": "Alice",
    "supply_donated": null,
    "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
    "class_period": 2,
    "last_name": "Fisher",
    "id": "5700228501995520",
    "self": "http://localhost:8080/5700228501995520"
}
```

Failure

Status: 400 Bad Request

```
{
    "Error": "The request object is missing at least one of the required attributes or
 the data provided is invalid."
}
```

Failure

Status: 401 Unauthorized

```
{
"Error":  "Missing or invalid JWT token."
}
```

Failure

Status: 403 Forbidden

```
{
    "Error": "Unauthorized"
}
```

Failure

Status: 404 Not Found

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

Patch a Student

Allows an authorized user to update a student– it is possible to update the following attributes: first_name, last_name, and class_period.

The property supply_donated can be changed using the /supplies/:supply_id/students/:student_id routes.

PATCH /students

Request

Path Parameters

Name	Description
student_id	ID of the student

Request Authorization

Required (Bearer Token)

Request Body Format

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
first_name	Student first name	No
last_name	Student last name	No
class_period	Student class period	No

Request Body Example

```
{
    "first_name": "Kym"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If the request does not include the bearer token authorization, the supply will not be created, and 401 status code will be returned.

Failure	403 Forbidden	If the request contains a bearer token, but it is not that of the teacher of the supply, then the 403 status code will be returned.
Failure	404 Not Found	No supply with this supply_id exists

Response Examples

Success

Status: 201 Created

```
{
    "num_needed": 10,
    "students": [],
    "description": "tissue box",
    "num_supplied": 0,
    "teacher_autho": "auth0|627c73d9eb7d06006f76a973",
    "id": "5751558595674112",
    "self": "http://localhost:8080/supplies/5751558595674112"
}
```

Failure

Status: 400 Bad Request

```
{
    "Error": "The request object is missing at least one of the required attributes or
 the data provided is invalid."
}
```

Failure

Status: 401 Unauthorized

```
{
"Error":  "Missing or invalid JWT token."
}
```

Failure

Status: 403 Forbidden

```
{
    "Error": "Unauthorized"
}
```

Failure

Status: 404 Not Found

```
{
"Error":  "No supply with this supply_id exists"
}
```

Student donates a Supply

An authorized teacher can create a relationship between a teacher's student and supply. In other words, a student can donate a supply. A supply can have many students – each student will donate this supply item. A student can donate at most one supply.

When the student_id is added to the students array for the Supply entity, the attribute num_supplied is incremented by 1.

PUT /supplies/:supply_id/students/:student_id

Request

Path Parameters

Name	Description
supply_id	ID of the supply
student_id	ID of the student

Request Authorization

Required (Bearer Token)

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a supply exists with this supply_id, a student exists with this student_id and the student doesn't already have a supply.
Failure	401 Unauthorized	If the request does not include the bearer token authorization, the supply will not be created, and 401 status code will be returned.
Failure	403 Forbidden	If the request contains a bearer token, but it is not that of the teacher of the supply, then the 403 status code will be returned.
Failure	404 Not Found	The specified supply and/or student does not exist

Response Examples

Success

Status: 204 No Content

Failure

```
Status: 401 Unauthorized
{
"Error":  "Missing or invalid JWT token."
}
```

Failure

```
Status: 403 Forbidden
{
    "Error": "Unauthorized"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "The specified supply and/or student does not exist"
}
```

Remove a relationship between student and supply

An authorized teacher can delete a relationship between a teacher's student and supply. In other words, a student can cancel the donation of a supply. Once a supply is removed from a student, the student's supply_donated is set to null.

When the student_id is removed from the students array for the Supply entity, the attribute num_supplied is decremented by 1.

```
DELETE /supplies/:supply_id/students/:student_id
```

Request

Path Parameters

Name	Description
supply_id	ID of the supply
student_id	ID of the student

Request Authorization

Required (Bearer Token)

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a supply exists with this supply_id, a student exists with this student_id and the student doesn't already have a supply.
Failure	401 Unauthorized	If the request does not include the bearer token authorization, the supply will not be created, and 401 status code will be returned.
Failure	403 Forbidden	If the request contains a bearer token, but it is not that of the teacher of the supply, then the 403 status code will be returned.
Failure	404 Not Found	The specified supply and/or student does not exist

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized
{
"Error":  "Missing or invalid JWT token."
}
```

Failure

```
Status: 403 Forbidden
{
    "Error": "Unauthorized"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "The specified supply and/or student does not exist"
}
```