Portfolio Assignment: API Spec

CS 493: Cloud Application Development

Spring 2021

Oregon State University

Last Update: June 6th, 2021

URL to Google Cloud App: https://cs493-portfolio-314720.wl.r.appspot.com/

Create a Boat (Protected)

Allows user to create a new boat.

```
POST /boats
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat.	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Integer	Length of the boat in feet.	Yes

Request Body Example

```
{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is empty, contains an unexpected attribute, is missing any of the 3 required attributes, more than 3 attributes are received, or one of the attributes does not meet the criteria below, the object will not be created and a 400 response will be returned. Valid name/type inputs: - Must be a string - Must contain 2-25 characters - Must start with a letter - Must end with a letter or number - May contain the following special characters: .’/- - May contain spaces

		Valid length inputs: - Must be an integer - Must be in range 3-2000
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	- Boat name already exists, or - This user does not have access to this resource.
Failure	406 Not Acceptable	Requested response type not supported
Failure	415 Unsupported Media Type	Request must be JSON

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The **self** attribute will contain the live link to the REST resource corresponding to this boat. In other words, this is the URL to get this newly created boat.
- The **owner** attribute will contain the ID number taken from the Google user's JWT 'sub' property.
- The **loads** attribute contains all loads carried on the boat. In the case of a newly created boat, it will always be an empty list.

Success

```
Status: 201 Created
{
  "id": 5650266825162752,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "110068366175927215356",
  "loads": [],
  "self": "http://127.0.0.1:8080/boats/5650266825162752"
}
```

Failures

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

```
Status: 401 Unauthorized
{
  "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
  "Error": "Boat name already exists"
}
```

```
Status: 406 Not Acceptable
{
   "Error": “Requested response type not supported”
}
```

```
Status: 415 Unsupported Media Type
{
   "Error": "Request must be in JSON"
}
```

Get a Boat (Protected)

Allows user to get an existing boat. Must be owned by user.

```
GET /boats/:boat_id
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	Type	Description
boat_id	Integer	ID of the boat

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	Requested response type not supported

Response Examples

Success

```
Status: 200 OK
{
  "id": 5650266825162752,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "110068366175927215356",
  "loads": [],
  "self": "http://127.0.0.1:8080/boats/5650266825162752"
}
```

Failures

```
Status: 401 Unauthorized
{
  "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 404 Not Found
{
  "Error":  "No boat with this boat_id exists"
}
```

```
Status: 406 Not Acceptable
{
   "Error": “Requested response type not supported”
}
```

Get all owned Boats with pagination (Protected)

Get all boats owned by the user making the request, 5 at a time using pagination.

```
GET /boats
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a 'Bearer' token.

Request

Path Parameters

Name	Type	Description
cursor	String	Cursor to navigate to 'next' page of results. Initial request does not contain a cursor.

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	406 Not Acceptable	Requested response type not supported

Response Examples

- Response will contain a "boats" attribute with a list of 5 boats. If there are more boats in the database, then the response will contain a "next" attribute, which will contain a URL to direct the user to the next page of results (using a cursor in the given URL). If there is no "next" attribute, then the end of the results has been reached.

Success

```
Status: 200 OK
{
   "boats": [
      {
         "id": 5644895230361600,
         "name": "Sea Witch 3",
         "type": "Catamaran 3",
         "length": 28,
         "owner": "110068366175927215356",
         "loads": [],
         "self": "http://127.0.0.1:8080/boats/5644895230361600"
      },
      {
         "id": 5650266825162752,
         "name": "Sea Witch",
         "type": "Catamaran",
         "length": 28,
         "owner": "110068366175927215356",
         "loads": [],
         "self": "http://127.0.0.1:8080/boats/5650266825162752"
      },
      {
         "id": 5652538258882560,
         "name": "Sea Witch 4",
         "type": "Catamaran 4",
         "length": 28,
         "owner": "110068366175927215356",
         "loads": [],
         "self": "http://127.0.0.1:8080/boats/5652538258882560"
      },
      {
         "id": 5676594102272000,
         "name": "Sea Witch 2",
         "type": "Catamaran 2",
         "length": 28,
         "owner": "110068366175927215356",
         "loads": [],
         "self": "http://127.0.0.1:8080/boats/5676594102272000"
      },
      {
         "id": 5689449946021888,
         "name": "Sea Witch 5",
         "type": "Catamaran 5",
         "length": 28,
         "owner": "110068366175927215356",
         "loads": [],
         "self": "http://127.0.0.1:8080/boats/5689449946021888"
      }
   ],
   "next":
"http://127.0.0.1:8080/boats?cursor=Ci4SKGoSc35jczQ5My1odzgtMzE0MTE3chILEgVib2F0cxiAgIDY5NCNCgwYACAA"
}
```

Failures

```
Status: 401 Unauthorized
{
   "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 406 Not Acceptable
{
   "Error": “Requested response type not supported”
}
```

Edit a Boat (PUT) (Protected)

Allows user to edit a boat. Must be owned by user and all attributes must be present in body.

```
PUT /boats/:boat_id
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	Type	Description
boat_id	Integer	ID of the boat

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat.	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Integer	Length of the boat in feet.	Yes

Request Body Example

```
{
  "name": "Sea Witch 1.2",
  "type": "Catamaran 1.2",
  "length": 30
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is empty, contains an unexpected attribute, is missing any of the 3 required attributes, more than 3 attributes are received, or one of the attributes does not meet the criteria below, the object will not be created and a 400 response will be returned. Valid name/type inputs: - Must be a string - Must contain 2-25 characters - Must start with a letter - Must end with a letter or number - May contain the following special characters: .’/-

		- May contain spaces Valid length inputs: - Must be an integer - Must be in range 3-2000
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	- Boat name already exists - This user does not have access to this resource.
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	Requested response type not supported
Failure	415 Unsupported Media Type	Request must be JSON

Response Examples

Success

```
Status: 200 OK
{
   "id": 5650266825162752,
   "name": "Sea Witch 1.2",
   "type": "Catamaran 1.2",
   "length": 30,
   "owner": "110068366175927215356",
   "loads": [],
   "self": "http://127.0.0.1:8080/boats/5650266825162752"
}
```

Failures

```
Status: 400 Bad Request
{
   "Error": "Invalid boat name"
}
```

```
Status: 401 Unauthorized
{
   "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 404 Not Found
{
"Error":  "No boat with this boat_id exists"
}
```

```
Status: 406 Not Acceptable
{
   "Error": “Requested response type not supported”
}
```

```
Status: 415 Unsupported Media Type
{
   "Error": "Request must be in JSON"
}
```

Edit a Boat (PATCH) (Protected)

Allows user to edit a boat. Must be owned by user.

```
PATCH /boats/:boat_id
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	**Type**	**Description**
boat_id	Integer	ID of the boat

Request Body

Required. The body must contain at least one attribute.

Request Body Format

JSON

Request JSON Attributes

Name	**Type**	**Description**	**Required?**
name	String	The name of the boat.	No
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	No
length	Integer	Length of the boat in feet.	No

Request Body Example

```
{
  "name": "Sea Witch 1.2",
  "type": "Catamaran 1.2",
  "length": 30
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	201 Created	
Failure	400 Bad Request	If the request is empty, contains an unexpected attribute, is missing any of the 3 required attributes, has more than 3 attributes are received, or one of the attributes does not meet the criteria below, the object will not be created and a 400 response will be returned. Valid name/type inputs: - Must be a string - Must contain 2-25 characters - Must start with a letter - Must end with a letter or number - May contain the following special characters: .’/-

		- May contain spaces Valid length inputs: - Must be an integer - Must be in range 3-2000
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	- Boat name already exists - This user does not have access to this resource.
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	Requested response type not supported
Failure	415 Unsupported Media Type	Request must be JSON

Response Examples

Success

```
Status: 200 OK
{
   "id": 5650266825162752,
   "name": "Sea Witch 1.2",
   "type": "Catamaran 1.2",
   "length": 30,
   "owner": "110068366175927215356",
   "loads": [],
   "self": "http://127.0.0.1:8080/boats/5650266825162752"
}
```

Failures

```
Status: 400 Bad Request
{
   "Error": "The request contains too many attributes."
}
```

```
Status: 401 Unauthorized
{
   "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "Boat name already exists"
}
```

```
Status: 404 Not Found
{
"Error":  "No boat with this boat_id exists"
}
```

```
Status: 406 Not Acceptable
{
   "Error": “Requested response type not supported”
}
```

```
Status: 415 Unsupported Media Type
{
   "Error": "Request must be in JSON"
}
```

Delete a Boat (Protected)

Allows you to delete a boat. Note that if the boat is currently carrying any loads, the loads will be unloaded and will not have a “carrier”.

```
DELETE /boats/:boat_id
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	Type	Description
boat_id	Integer	ID of the boat

Request Body

None

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failures

```
Status: 401 Unauthorized
{
   "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 404 Not Found
{
"Error":  "No boat with this boat_id exists"
}
```

Create a Load (Protected)

Allows you to create a new Load.

```
POST /loads
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
volume	Integer	The volume of the load.	Yes
content	String	The contents of the load.	Yes

Request Body Example

```
{
  "volume": 1,
  "content": "Bananas1"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is empty, missing the volume or the content attribute, contains an unexpected attribute, has more than two attributes, or the attributes do not meet the criteria below, the load must not be created, and a 400 status code must be returned. Valid volume input: - Must be integer between 1 and 10,000 Valid content: - Must be string between 3 and 50 characters
Failure	401 Unauthorized	Invalid JWT
Failure	406 Not Acceptable	Requested response type not supported
Failure	415 Unsupported Media Type	Request must be in JSON

Response Examples

- The value of the attribute **carrier** is the ID of the boat currently carrying the load. If the load is not assigned to a boat, the value of **carrier** should be null.
- The value of the attribute **creation_date** is the date that the load was created (the current date, determined by the API), in YYYY-MM-DD format. This date is stored in the database.
- The value of the attribute **self** is a live link to the REST resource corresponding to this load. In other words, this is the URL to get this newly created load.
- The value of the attribute **owner** will contain the ID number taken from the Google user's JWT 'sub' property.

Success

```
Status: 201 Created
{
  "id": 5081054809423872,
  "volume": 1,
  "carrier": null,
  "content": "Bananas1",
  "creation_date": "2021-06-01",
  "owner": "110068366175927215356",
  "self": "http://127.0.0.1:8080/loads/5081054809423872"
}
```

Failures

```
Status: 400 Bad Request
{
  "Error": "The request body must not be empty."
}
```

```
Status: 401 Unauthorized
{
  "Error": "Invalid JWT"
}
```

```
Status: 406 Not Acceptable
{
  "Error": “Requested response type not supported”
}
```

```
Status: 415 Unsupported Media Type
{
  "Error": "Request must be in JSON"
}
```

Get a Load (Protected)

Allows user to get an existing load. Must be owned by user.

```
GET /loads/:load_id
```

Request

Path Parameters

Name	Type	Description
load_id	Integer	ID of the load

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	404 Not Found	No load with this load_id exists
Failure	406 Not Acceptable	Requested response type not supported

Response Examples

Success

```
Status: 200 OK
{
  "id": 5081054809423872,
  "volume": 1,
  "carrier": {
    "id": 5644004762845184,
    "name": "Sea Witch",
    "self": "http://127.0.0.1:8080/boats/5644004762845184"
  },
  "content": "Bananas1",
  "creation_date": "2021-06-01",
  "owner": "110068366175927215356",
  "self": "http://127.0.0.1:8080/loads/5081054809423872"
}
```

Failures

```
Status: 401 Unauthorized
{
  "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 404 Not Found
{
"Error":  "No load with this load_id exists"
}
```

```
Status: 406 Not Acceptable
{
   "Error": “Requested response type not supported”
}
```

Get all user Loads with pagination (Protected)

Get all the user’s loads, 5 per page with pagination.

```
GET /loads
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	Type	Description
cursor	String	Cursor to navigate to ‘next’ page of results. Initial request does not contain a cursor.

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	406 Not Acceptable	Requested response type not supported.

Response Examples

- Response will contain a **loads** attribute with a list of up to 5 loads. If there are more loads in the database, then the response will contain a **next** attribute, which will contain a URL to direct the user to the next page of results (using a cursor in the given URL). If there is no **next** attribute, then the end of the results has been reached.

Success

```
Status: 200 OK
{
  "loads": [
    {
      "id": 4789736740749312,
      "volume": 5,
      "carrier": null,
      "content": "Bananas5",
      "creation_date": "2021-06-01",
      "owner": "110068366175927215356",
      "self": "http://127.0.0.1:8080/loads/4789736740749312"
    },
    {
      "id": 4797943718608896,
      "volume": 4,
      "carrier": null,
      "content": "Bananas4",
      "creation_date": "2021-06-01",
      "owner": "110068366175927215356",
      "self": "http://127.0.0.1:8080/loads/4797943718608896"
    },
    {
      "id": 5069549128908800,
      "volume": 6,
      "carrier": null,
      "content": "Bananas6",
      "creation_date": "2021-06-01",
      "owner": "110068366175927215356",
      "self": "http://127.0.0.1:8080/loads/5069549128908800"
    },
    {
      "id": 5081054809423872,
      "volume": 1,
      "carrier": {
        "id": 5644004762845184,
        "name": "Sea Witch",
        "self": "http://127.0.0.1:8080/boats/5644004762845184"
      },
      "content": "Bananas1",
      "creation_date": "2021-06-01",
      "owner": "110068366175927215356",
      "self": "http://127.0.0.1:8080/loads/5081054809423872"
    },
    {
      "id": 6205318602162176,
      "volume": 3,
      "carrier": null,
      "content": "Bananas3",
      "creation_date": "2021-06-01",
      "owner": "110068366175927215356",
      "self": "http://127.0.0.1:8080/loads/6205318602162176"
    }
  ],
  "next":
"http://127.0.0.1:8080/loads?cursor=CjQSLmoYbX5jczQ5My1wb3J0Zm9saW8tMzE0NzIwchILEgVsb2FkcxiAgICYxfaCCwwYACAA"
}
```

Failures

```
Status: 401 Unauthorized
{
   "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 406 Not Acceptable
{
   "Error": “Requested response type not supported”
}
```

Edit a Load (PUT) (Protected)

Allows user to edit a load. Must be owned by user and all attributes must be present in body.

```
PUT /loads/:load_id
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	Type	Description
load_id	Integer	ID of the load

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
volume	Integer	The volume of the load in cubic feet.	Yes
content	String	The content of the load.	Yes

Request Body Example

```
{
  "volume": 10,
  "content": "Helicopters"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is empty, missing the volume or the content attribute, contains an unexpected attribute, has more than two attributes, or the attributes do not meet the criteria below, the load must not be created, and a 400 status code must be returned. Valid volume input: - Must be integer between 1 and 10,000 Valid content: Must be string between 3 and 50 characters
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	404 Not Found	No load with this load_id exists
Failure	406 Not Acceptable	Requested response type not supported

Failure	415 Unsupported Media Type	Request must be JSON

Response Examples

Success

```
Status: 200 OK
{
   "id": 5650266825162752,
   "name": "Sea Witch 1.2",
   "type": "Catamaran 1.2",
   "length": 30,
   "owner": "110068366175927215356",
   "loads": [],
   "self": "http://127.0.0.1:8080/boats/5650266825162752"
}
```

Failures

```
Status: 400 Bad Request
{
   "Error": "Unexpected attribute in request."
}
```

```
Status: 401 Unauthorized
{
   "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 404 Not Found
{
"Error":  " No load with this load_id exists "
}
```

```
Status: 406 Not Acceptable
{
   "Error": “Requested response type not supported”
}
```

```
Status: 415 Unsupported Media Type
{
   "Error": "Request must be in JSON"
}
```

Edit a Load (PATCH) (Protected)

Allows user to edit a load. Must be owned by user.

```
PATCH /loads/:load_id
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	Type	Description
load_id	Integer	ID of the load

Request Body

Required. The body must contain at least one attribute.

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
volume	Integer	The volume of the load in cubic feet.	No
content	String	The content of the load.	No

Request Body Example

```
{
  "name": "Sea Witch 1.2",
  "type": "Catamaran 1.2",
  "length": 30
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is empty, missing the volume or the content attribute, contains an unexpected attribute, has more than two attributes, or the attributes do not meet the criteria below, the load must not be created, and a 400 status code must be returned. Valid volume input: - Must be integer between 1 and 10,000 Valid content: - Must be string between 3 and 50 characters
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	404 Not Found	No load with this load_id exists

Failure	406 Not Acceptable	Requested response type not supported
Failure	415 Unsupported Media Type	Request must be in JSON

Response Examples

Success

```
Status: 200 OK
{
   "id": 5650266825162752,
   "name": "Sea Witch 1.2",
   "type": "Catamaran 1.2",
   "length": 30,
   "owner": "110068366175927215356",
   "loads": [],
   "self": "http://127.0.0.1:8080/boats/5650266825162752"
}
```

Failures

```
Status: 400 Bad Request
{
   "Error": "Invalid load volume"
}
```

```
Status: 401 Unauthorized
{
   "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 404 Not Found
{
"Error":  "No load with this load_id exists "
}
```

```
Status: 406 Not Acceptable
{
   "Error": “Requested response type not supported”
}
```

```
Status: 415 Unsupported Media Type
{
   "Error": "Request must be in JSON"
}
```

Delete a Load (Protected)

Allows you to delete a load. If the load being deleted is assigned to a boat, the load is removed from the boat.

```
DELETE /loads/:load_id
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	**Type**	**Description**
load_id	Integer	ID of the load

Request Body

None

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	204 No Content	
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	404 Not Found	No load with this load_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failures

```
Status: 401 Unauthorized
{
   "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 404 Not Found
{
"Error":  "No load with this load_id exists"
}
```

Assign Load to Boat (Protected)

Assign Load to a Boat. Boat must be owned by user. A load cannot be assigned to multiple boats.

```
PUT /boats/:boat_id/loads/:load_id
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	Type	Description
load_id	Integer	ID of the load
boat_id	Integer	ID of the boat

Request Body

None

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with this load_id, and the load is not assigned to a boat.
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	- This load is already assigned to a boat - This user doesn’t have access to this resource.
Failure	404 Not Found	- No boat with this boat_id exists - No load with this load_id exists - Neither the boat nor load with the given IDs exist

Response Examples

Success

```
Status: 204 No Content
```

Failures

```
Status: 401 Unauthorized
{
  "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
  "Error": "This load is already assigned to a boat"
}
```

```
Status: 404 Not Found
{
  "Error": "Neither the boat nor load with the given IDs exist"
}
```

Remove Load from Boat (Protected)

Remove Load from a boat. Boat must be owned by user. The load now has no carrier.

```
DELETE /boats/:boat_id/loads/:load_id
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	**Type**	**Description**
load_id	Integer	ID of the load
boat_id	Integer	ID of the boat

Request Body

None

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with this load_id, and this load is on this boat.
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	404 Not Found	- No boat with this boat_id exists - No load with this load_id exists - Neither the boat nor load with the given IDs exist - No load with this load_id is on the boat with this boat_id

Response Examples

Success

```
Status: 204 No Content
```

Failures

```
Status: 401 Unauthorized
{
  "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
  "Error": "This user does not have access to this resource."
}
```

```
Status: 404 Not Found
{
  "Error": “No load with this load_id is on the boat with this boat_id "
}
```

Get all Loads for a given Boat (Protected)

Get all Loads assigned to a given boat. Boat must be owned by user. No pagination.

```
Get /boats/:boat_id/loads
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	Type	Description
boat_id	Integer	ID of the boat

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	404 Not Found	No boat with this boat_id exists.
Failure	406 Not Acceptable	Requested response type not supported

Response Examples

Success

```
Status: 200 OK
[
  {
    "id": 5081054809423872,
    "volume": 1,
    "carrier": {
      "id": 5644004762845184,
      "name": "Sea Witch",
      "self": "http://127.0.0.1:8080/boats/5644004762845184"
    },
    "content": "Bananas1",
    "creation_date": "2021-06-01",
    "owner": "110068366175927215356",
    "self": "http://127.0.0.1:8080/loads/5081054809423872"
  }
]
```

Failures

```
Status: 401 Unauthorized
{
   "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 404 Not Found
{
   "Error": " No boat with this boat_id exists"
}
```

```
Status: 406
{
   "Error": “Requested response type not supported”
}
```

Get a User (Protected)

Allows user to get their own user information. Must be the user making the request.

```
GET /users/:user_id
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a ‘Bearer’ token.

Request

Path Parameters

Name	Type	Description
user_id	Integer	ID of the user

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	404 Not Found	No user with this user_id exists
Failure	406 Not Acceptable	Requested response type not supported

Response Examples

Success

```
Status: 200 OK
{
  "id": "12345",
  "given_name": "Dyllan Edward",
  "family_name": "Vangemert",
  "boats": [
    {
      "id": 5710353417633792,
      "self": "http://127.0.0.1:8080/boats/5710353417633792"
    }
  ],
  "loads": [],
  "self": "http://127.0.0.1:8080/users/12345"
}
```

Failures

```
Status: 401 Unauthorized
{
  "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 404 Not Found
{
  "Error":  "No user with this user_id exists"
}
```

```
Status: 406 Not Acceptable
{
   "Error": “Requested response type not supported”
}
```

Delete a User (Protected)

Allows user to delete their own user information. All the boats and loads owned by the user will be deleted. Note that all loads being carried by any deleted boat will be updated, with the load's 'carrier' attribute changed to null. Must be the user making the request.

```
DELETE /users/:user_id
```

Authorization

Google OAuth login Json Web Token (JWT) required in Authorization header as a 'Bearer' token.

Request

Path Parameters

Name	Type	Description
user_id	Integer	ID of the user

Request Body

None

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Invalid JWT
Failure	403 Forbidden	This user does not have access to this resource.
Failure	404 Not Found	No user with this user_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failures

```
Status: 401 Unauthorized
{
   "Error": "Invalid JWT"
}
```

```
Status: 403 Forbidden
{
   "Error": "This user does not have access to this resource."
}
```

```
Status: 404 Not Found
{
"Error":  "No user with this user_id exists"
}
```

Get all Users (Unprotected)

Get all users. No pagination.

```
GET /users
```

Authorization

None

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	200 OK	
Failure	406 Not Acceptable	Requested response type not supported

Response Examples

Success

```
Status: 200 OK
{
   "users": [
      {
         "id": "12345",
         "given_name": "Dyllan Edward",
         "family_name": "Vangemert",
         "boats": [
            {
               "id": 5710353417633792,
               "self": "http://127.0.0.1:8080/boats/5710353417633792"
            }
         ],
         "loads": [],
         "self": "http://127.0.0.1:8080/users/12345"
      }
   ]
}
```

Failures

```
Status: 406
{
   "Error": “Requested response type not supported”
}
```

Supported/Unsupported Routes

The following table shows all API endpoints and which request methods they support/don’t support. All other request methods are not supported by the API.

	GET	POST	PUT	PATCH	DELETE
/boats	Y	Y	N	N	N
/boats/<boat_id>	Y	N	Y	Y	Y
/loads	Y	Y	N	N	N
/loads/<load_id>	Y	N	Y	Y	Y
/boats/<boat_id>/loads	Y	N	N	N	N
/boats/<boat_id>/loads/<loads_id>	Y	N	Y	N	Y
/users	Y	N	N	N	N
/users/<user_id>	Y	N	N	N	Y

Y = Route supported N = Route unsupported

Datastore Model

Boat

Property Name	**Notes**	**Required**
id	Integer. The id of the boat. Datastore automatically generates it.	*
name	String. Name of the boat.	Yes
type	String. Type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Integer. The length of the boat in feet.	Yes
owner	Integer. User's id for user who created the boat. Generated by server.	Yes (handled by server, user has no control over this)

Load

Property Name	**Notes**	**Required**
id	Integer. The id of the load. Datastore automatically generates it.	*
volume	Integer. The volume of the load in cubic feet.	Yes
carrier	Integer. If the load is on a boat, then carrier is the id of that boat. Otherwise null.	Yes
creation_date	String. Date the load was created. Generated by server.	Yes (handled by server, user has no control over this)
owner	Integer. User's id for user who created the load. Generated by server.	Yes (handled by server, user has no control over this)

User

Property Name	**Notes**	**Required**
id	Integer. The id of the user, taken from the "sub" property in in the given JWT.	Yes (handled by OAuth flow, no user input needed)
given_name	Integer. The volume of the load in cubic feet.	Yes (handled by OAuth flow, no user input needed)
family_name	If the load is on a boat, then carrier is the id of that boat. Otherwise null.	Yes (handled by OAuth flow, no user input needed)

Comments

- Boat: The return attributes of 'self' and 'loads' are not stored in Datastore. Both attributes are generated 'on the fly'. 'self' is generated using the basic URL structure, with the boat's id tacked onto the end. 'loads' is initialized as an empty list, then a query is made for all loads with a "carrier" id matching the boat_id. The response loads are then added to the list of loads and formatted as demonstrated in the response examples.

- Load: The “carrier” attribute is stored as an id only (corresponding to the respective boat carrying the load). Upon getting the load, a query is made to get the boat with the matching id stored in “carrier”. The “carrier” attribute is then updated to contain an object representing an abbreviated representation of the boat (‘id’, ‘name’, and ‘self’ only).

- User: The return attributes ‘self, ‘boats’, and ‘loads’ are not stored in Datastore. All attributes are generated on the fly. ‘self’ is generated using the basic URL structure, with the user’s id at the end. ‘boats’ is initialized as an empty list, then a query is made for all boats with the ‘owner’ id matching the user_id. The response boats are then added to the list of boats and formatted to demonstrate the examples. The same applies for ‘loads’.

- SIDENOTE: API time is based on PST time zone.