RESTful API Project

jdickinson214

Google URL: https://dickinsj-project.uc.r.appspot.com/

Built using Python3 and Flask on Google Cloud Platform

The link above is also the starting point for logging in. User is taken to a site with a hyper link that takes them through the OAuth2.0 process, first asking them to log into their google account to generate a JWT. This JWT is then displayed on the screen for the user. There is then a verify link to test it and see if it can retrieve the user's email address.

Data Models:

	User	Boat	Load
Authorization to view one?	Yes	Yes	No
Authorization to view all?	No	Yes	No

Users to Boats: one to many relationship.

Loads to Boats: one to many relationship.

Users to Loads: No direct relationship, but transitively one to many through boats entity.

User	
uniqueID	User's email, String, "example@gmail.com"
Id	Database generated ID #, String, "12345678910"

User's uniqueID is store in a database within google along with an ID whenever they first go through the OAuth2.0 process and generate their first JWT. The JWTs are not store and expire after a short time.

The JWT is used as the Bearer token for each request regarding boats and getting a specific user.

Notes: if using a method type not allowed (ex. PUT /users), you will get a 405 status code.

If request body or Header Content-Type is not application/json, you will get a 415 status.

RESTful API Spec Sheet

Last Update: June 8, 2020

Get a User

Allows you to get this user. Authorization Required.

GET /users/:user_id

Request

Request Parameters

Name	Type	Description	Required?
user_id	String	ID of the user	Yes
Authorization Bearer Token	Header	User's JWT as Bearer token. Ex. {"Authorization": "Bearer 12938192863198236"}	Yes

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	User's JWT is invalid or expired
Failure	403 Forbidden	User's JWT is not the user they are requesting to view
Failure	404 Not Found	JWT valid but no entity with this id exists
Failure	406	Accept header doesn't include application/json

Response Examples

Success

```
Status: 200 OK
{
  "user": {
     "uniqueID": "example@gmail.com",
     "id": "570002138789273490",
     "boats": [
      {
        "id": "abc123",
        "name": "Sea Witch",
        "type": "Catamaran",
        "length": 28,
        "loads": [
          { "id": "aaa111",
            "content": "Stuff",
            "self": "https://<your-app>/loads/aaa111"}
        ],
        "self": "https://<your-app>/boats/abc123"
}]}}
```

Failure

Status: 401 Unauthorized
 	
{
“Error”: “Missing or invalid JWT”
}

Status: 403 Forbidden
 	
{
“Error”: “User not authorized to access this content”
}

Status: 404 Not Found
 	
{
“Error”: “JWT valid but no entity with this id exists”
}

Note: all future 401, 403, and 404 responses will look identical to these, therefore they will not be repeated as examples further on.

List all Users

List all users. Authorization NOT required.

GET /users

Request

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406	Accept header doesn't include application/json

Response Examples

Success

Status: 200 OK


```
{
  "users": [
    {
      "uniqueID": "example1@gmail.com",
      "id": "654321"
    },
    {
      "uniqueID": "example2@gmail.com",
      "id": "123456"
    }
  ]
```

Create a Boat

Allows you to create a new boat.

POST /boats

Request

Request Parameters

Name	Type	Description	Required?
Authorization Bearer Token	Header	User's JWT as Bearer token. Ex. {"Authorization": "Bearer 12938192863198236"}	Yes

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
loads	Array	Array of loads that are loaded onto this boat. Defaults to empty. It advised to create a load separately, and then 'load' the load onto the boat using the correct endpoint detailed below.	No
Owner	String	The 'uniqueID' or email of the user that created this boat.	No*

*Technically it is required, but the value is obtained automatically and does not need to be entered by the user anywhere.

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 missing any of the 3 required attributes, the boat will not be created, and 400 status code will return.
Failure	401 Unauthorized	User's JWT is invalid or expired
Failure	403 Forbidden	User's JWT is not the owner of the boat

Failure	404 Not Found	JWT valid but no entity with this id exists
Failure	406	Accept header doesn't include application/json
Failure	415	Content-Type is not application/json.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The 'self' attribute is not stored on the database. It contains the live link to the REST resource corresponding to this boat.

Success

Status: 201 Created


```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": example@gmail.com,
  "loads": [],
  "self": "https://dickinsj-project.uc.r.appspot.com/boats/abc123"
}
```


Failure

Status: 400 Bad Request


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

PUT/PATCH a Boat

Allows you to create a new load. Authorization required.

```
PUT    /boats/<boat_id>
PATCH  /boats/<boat_id>
```

Request

Request Parameters

Name	Type	Description	Required?
boat_id	String	ID of the boat	Yes
Authorization Bearer Token	Header	User's JWT as Bearer token. Ex. {"Authorization": "Bearer 12938192863198236"}	Yes

Request Body

Required

Request Body Format

JSON

Request JSON Attributes (At least one is required)

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
loads	Array	Array of loads that are loaded onto this boat. Defaults to empty. It advised to create a load separately, and then 'load' the load onto the boat using the correct endpoint detailed below.	No

Request Body Example

```
{
  "name": "Jenny2"
  "length": "33"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 Created	
Failure	400 Bad Request	If the request has an invalid attribute.
Failure	401 Unauthorized	User's JWT is invalid or expired
Failure	403 Forbidden	User's JWT is not the owner of the boat
Failure	404 Not Found	JWT valid but no entity with this id exists
Failure	406	Accept header doesn't include application/json
Failure	415	Content-Type is not application/json.

Note: PUT not retain un-mentioned attributes (weight, content, delivery_date) and will set these to null. PATCH will retain any un-mentioned attributes previous values.

Success

Status: 200 Ok

PATCH


```
{
  “id”: “abc123”,
  “name”: “Jenny2”,
  “type”: “Catamaran”,
  “length”: 33,
  “owner”: example@gmail.com,
  “loads”: [],
  “self”: “https://dickinsj-project.uc.r.appspot.com/boats/abc123”
}
```


PUT


```
{
  “id”: “abc123”,
  “name”: “Jenny2”,
  “type”: null,
  “length”: 33,
  “owner”: example@gmail.com,
  “loads”: [],
  “self”: “https://dickinsj-project.uc.r.appspot.com/boats/abc123”
}
```

Get a Boat

Allows you to get an existing boat. Authorization required.

GET /boats/:boat_id

Request

Request Parameters

Name	Type	Description	Required?
boat_id	String	ID of the boat	Yes
Authorization Bearer Token	Header	User's JWT as Bearer token. Ex. {"Authorization": "Bearer 12938192863198236"}	Yes

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	User's JWT is invalid or expired
Failure	403 Forbidden	User's JWT is not the owner of the boat
Failure	404 Not Found	JWT valid but no entity with this id exists
Failure	406	Accept header doesn't include application/json

Response Examples

Success

Status: 200 OK


```
{
  "type": "Yacht",
  "length": 99,
  "owner": "example@gmail.com",
  "name": "Odyssey",
  "loads": [],
  "id": "123456",
  "self": "https://dickinsj-project.uc.r.appspot.com/boats/123456"
}
```

List all Boats

List all the boats for associated JWT. Authorization Required.

GET /boats

Request

Request Parameters

Name	Type	Description	Required?
Authorization Bearer Token	Header	User's JWT as Bearer token. Ex. {"Authorization": "Bearer 12938192863198236"}	Yes

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	User's JWT is invalid or expired
Failure	406	Accept header doesn't include application/json

Response Examples

Success


```
Status: 200 OK
{
  "boats": [
    {
      "name": "Odyssey",
      "loads": [],
      "type": "Yacht",
      "length": 99,
      "owner": "example@gmail.com",
      "id": "5647975057457152",
      "self": "https://dickinsj-project.uc.r.appspot.com/boats/5647975057457152"
    },
    {
      "name": "Odyssey",
      "loads": [],
      "type": "Yacht",
      "length": 99,
      "owner": "example@gmail.com",
      "id": "5689540979195904",
      "self": "https://dickinsj-project.uc.r.appspot.com/boats/5689540979195904"
    },
    {
```

```
        "loads": [],
        "type": "Yacht",
        "length": 99,
        "owner": "example@gmail.com",
        "name": "Odyssey",
        "id": "5715110588841984",
        "self": "https://dickinsj-project.uc.r.appspot.com/boats/5715110588841984"
      },
      {
        "name": "Odyssey",
        "loads": [],
        "type": "Yacht",
        "length": 99,
        "owner": "example@gmail.com",
        "id": "5723707007827968",
        "self": "https://dickinsj-project.uc.r.appspot.com/boats/5723707007827968"
      },
      {
        "type": "Yacht",
        "length": 99,
        "owner": "example@gmail.com",
        "name": "Odyssey",
        "loads": [],
        "id": "5741714799067136",
        "self": "https://dickinsj-project.uc.r.appspot.com/boats/5741714799067136"
      }
    ],
    "next": "https://dickinsj-project.uc.r.appspot.com/boats?limit=5&offset=5"
}
```


Note: results are paginated to 5 per page. next is the link for the next set of results, also boats shown are the boats for this user. This endpoint does not show all boat in the system.

Delete a Boat

Allows you to delete a boat. Note whatever loads are currently on the boat, will be offloaded and the load's 'carrier' value is set to null. Authorization required.

DELETE /boats/:boat_id

Request

Request Parameters

Name	Type	Description	Required?
boat_id	String	ID of the boat	Yes
Authorization Bearer Token	Header	User's JWT as Bearer token. Ex. {"Authorization": "Bearer 12938192863198236"}	Yes

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	User's JWT is invalid or expired
Failure	403 Forbidden	User's JWT is not the owner of the boat
Failure	404 Not Found	JWT valid but no entity with this id exists

Response Examples

Success

Status: 204 No Content

Create a Load

Allows you to create a new load. Authorization NOT required.

POST /loads

Request

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
weight	Integer	The name of the boat.	Yes
content	String	The content of the load.	Yes
delivery_date	String	The date the load is to be delivered.	Yes
carrier	JSON	The boat carrying the load. Default is blank.	No

Request Body Example

```
{
  “weight”: 7,
  “content”: “Stuff”,
  “delivery_date”: “01/01/2021”
}
```

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 three attributes, the load will not be created, and 400 status code will return.
Failure	406	Accept header doesn’t include application/json
Failure	415	Content-Type is not application/json.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The value of the attribute carrier contains the ID and name of the boat currently at this load. If there is no boat at this load, the value of ‘carrier’ is null.
- The value of the attribute self is a live link to the REST resource corresponding to this load. Value is not stored in Datastore.

Success

Status: 201 Created


```
{
   “id”: “123abc”,
   “weight”: 7,
   “content”: “Stuff”,
   “delivery_date”: “01/01/2021”,
   “carrier”: null,
   “self”: “https://<your-app>/loads/123abc”
}
```


Failure

Status: 400 Bad Request


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

PUT/PATCH a Load

Allows you to create a new load. Authorization NOT required.

PUT /loads/<load_id> PATCH /loads/<load_id>

Request

Request Parameters

Name	Type	Description	Required?
load_id	String	ID of the load	Yes

Request Body

Required

Request Body Format

JSON

Request JSON Attributes (At least one is required)

Name	Type	Description	Required?
weight	Integer	The name of the boat.	Yes
content	String	The content of the load.	Yes
delivery_date	String	The date the load is to be delivered.	Yes
carrier	JSON	The boat carrying the load. Default is blank.	No

Request Body Example

```
{
  “weight”: 7,
  “delivery_date”: “01/01/2021”
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 Created	
Failure	400 Bad Request	If the request has an invalid attribute.
Failure	406	Accept header doesn’t include application/json
Failure	415	Content-Type is not application/json.

Note: PUT not retain un-mentioned attributes (weight, content, delivery_date) and will set these to null. PATCH will retain any un-mentioned attributes previous values.

Success

```
Status: 200 Ok

PUT
{
  “id”: “123abc”,
```

```
   “weight”: 7,
   “content”: null,
   “delivery_date”: “01/01/2021”,
   “carrier”: null,
   “self”: “https://<your-app>/loads/123abc”
}

PATCH
{
   “id”: “123abc”,
   “weight”: 7,
   “content”: bananas,
   “delivery_date”: “01/01/2021”,
   “carrier”: null,
   “self”: “https://<your-app>/loads/123abc”
}
```

Get a Load

Allows you to get an existing load. Authorization NOT required.

GET /loads/:load_id

Request

Request Parameters

Name	Type	Description	Required?
load_id	String	ID of the load	Yes

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No load with this load_id exists
Failure	406	Accept header doesn't include application/json

Response Examples

Success

Status: 200 OK

```
{
  "id": "aaa111",
  "weight": 7,
  "content": "Stuff",
  "delivery_date": "01/01/2021",
  "carrier": {
    "id": "abc321",
    "name": "Sea Witch",
    "self": "http://<your-app>/boats/abc123"},
  "self": "https://<your-app>/loads/aaa111"
}
```

Failure

Status: 404 Not Found

```
{
"Error": "No load with this load_id exists"
}
```

List all Loads

List all the loads. Authorization NOT required.

GET /loads

Request

Request Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406	Accept header doesn't include application/json

Response Examples

Success

Status: 200 OK


```
{[
  {
    "id": "aaa111",
    "content": "Stuff",
    "weight": 7,
     "carrier": [
       {"id": "abc123",
       "name": "Sea Witch",
       "self": "https://<your-app>/boats/ abc123"}
     ],
    "self": "https://<your-app>/loads/ aaa111"
  },
  {
    "id": "bbb222",
    "content": "Small Stuff",
    "weight": 3,
      {"id": "def456",
       "name": "Adventure",
       "self": "https://<your-app>/boats/ def456"}
     ],
    "self": "https://<your-app>/ loads / bbb222"
  },
```

```
  {
    “id”: “ccc333”,
    “content”: “Big Stuff”,
    “weight”: 100,
      {“id”: “xyz123”,
       “content”: “Hocus Pocus”,
       “self”: “https://<your-app>/boats/ xyz123”}
     ],
    “self”: “https://<your-app>/loads/ ccc333”
  }
],
“next”: “https://<your-app>/loads?limit=3/&offset=3”    #note: results are paginated to 3 per page
}
```

Delete a Load

Allows you to delete a load. If the load being deleted is on a boat, the load is removed from the boat, then deleted. Authorization NOT required.

DELETE /loads/:load_id

Request

Request Parameters

Name	Type	Description	Required?
load_id	String	ID of the load	Yes

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No load with this load_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
“Error”: “No load with this load_id exists”
}
```

Load put onto a Boat

Load is loaded onto a boat. Load contains reference to boat in 'carrier' field and Boat contains reference to load in 'loads' field. Authorization Required (User must own boat).

PUT /boats/:boat_id/loads/:load_id

Request

Request Parameters

Name	Type	Description	Required?
boat_id	String	ID of the boat	Yes
load_id	String	ID of the load	Yes

Request Body

None

Note: Set Content-Length to 0 in your request when calling out to this endpoint.

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 boat exists with this boat_id, a load exists with this load_id and the load is empty.
Failure	401 Unauthorized	User's JWT is invalid or expired
Failure	403 Forbidden	User's JWT is not the owner of the boat
Failure	403 Forbidden	Load is already on another boad
Failure	404 Not Found	JWT valid but no entity with this one of these IDs exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 403 Forbidden
 	
{
"Error": "The load is already loaded on a boat"
}

Comment

- A load cannot be assigned to multiple boats, but multiple loads can be on one boat.

Load is taken off a boat.

Load is 'offloaded' from a boat. The carrier is changed to null and the load is removed from the boat's 'loads'. Authorization Required (User must own boat).

DELETE boats/:boat_id/loads/:load_id

Request

Request Parameters

Name	Type	Description	Required?
boat_id	String	ID of the boat	Yes
load_id	String	ID of the load	Yes

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 boat exists with this boat_id, a load exists with this load_id and this load is on this boat.
Failure	401 Unauthorized	User's JWT is invalid or expired
Failure	403 Forbidden	User's JWT is not the owner of the boat
Failure	404 Not Found	JWT valid but no entity with this one of these IDs exists

Response Examples

Success

Status: 204 No Content