Assignment 9: API Doc

URL for my application: https://assignment-9-magadia.uc.r.appspot.com

URL for account creation/login: https://assignment-9-magadia.uc.r.appspot.com

Data Model

The app stores 3 kinds of entities in Datastore: Users, Boats, and Loads.
There is a one-to-many relationship between Boats and Loads. A Boat can have many loads, or none. A Load can only be assigned to one Boat, or to no boats. This relationship is modeled by storing the id of a Boat in the "carrier" property of a Load if that Load is assigned to that Boat. If the Load is unassigned, the carrier property remains null. Additionally, the User entity has a one-to-many relationship with Boats. A User can have multiple boats, or no boats. There is no direct relationship between Users and Loads, however. The load is a non-user entity and the boat is a user entity. All of a User's boats are protected – only the user can access and manipulate their boats directly. However, loads are not protected – no authorization or authentication is required to access and manipulate loads directly.

In my app, the 'sub' value in a user's JWT represents the unique identifier for that user. This value maps to a user's user_id, their unique identifier, in the app.
If a request is a protected URL and it needs to supply a user identifier, this must be done by setting the Authorization header for the request with a bearer token which is set to the user's JWT. The authorization header must be set in the following way: 'Bearer {{jwt}}'

Users:

Property	Data type	Notes	Required
id	integer	The user's id. Datastore generates it automatically and it is stored in the entity's key.	yes
user_id	string	This is the user_id associated with the user, and this is what shows up in the "owner" property of boats. It is the 'sub' value obtained from their JWT. This is considered the unique identifier for a user in my app.	yes

Boats:

Property	Data type	Notes	Required
id	integer	The boat's id. Datastore generates it automatically and it is stored in the entity's key.	yes

name	string	Name of the boat	yes
type	string	Type of the boat (yacht, for example)	yes
length	integer	Represents the length of the boat in feet	yes
owner	string	The user_id of the user who the boat belongs to.	yes

Note: The data types are not enforced by the app, so any values entered by the user for name, type, and length would be considered valid by the app. However, all of these properties are required for a Boat entity.

Loads:

Property	Data type	Notes	Required
id	integer	The load's id. Datastore generates it automatically and it is stored in the entity's key.	yes
quantity	integer	The quantity of the item specified in content.	yes
carrier	integer	If the load is assigned to a boat, this will store that boat's id. Otherwise, it is null.	yes
content	string	Represents the contents of the load (pencils, Legos, etc.)	yes
creation_date	string	The date the load was created in MM/DD/YYYY format. The app will generate this and store this by itself.	yes

Note: The data types are not enforced by the app, so any values entered by the user for content and quantity would be accepted by the app. However, all of these properties are required for a Load entity.

Endpoint Specifications

Create a boat

MIME Types of Request Body Accepted: application/json
MIME Type of Response: application/json

Request	POST /boats
Method	POST
URL	`https://assignment-9-magadia.uc.r.appspot.com/boats`
Path Parameters	None

Request Body	Required
Request Body Format	JSON
Protected	Yes

Request JSON Attributes:

Name	Description	Required?
name	Name of boat	Yes
type	Type of boat (ex. Sailboat)	Yes
length	Length of boat in feet	Yes

Example Request Body

```
{
    "name": "Sea Turtle2",
    "type": "Sailboat",
    "length": 187
}
```

Response Body Format: JSON

Outcome	Status Code	Notes
Success	201	
Failure	400	This is returned if any of the 3 attributes are missing. Also, the values of any included attributes are not validated for data type. They are assumed to be valid.
Failure	401	If invalid or missing JWT is sent.
Failure	406	If the request does not accept application/json MIME type as a response.

Example Responses:

Success: Status 201

```
{
    "id": 5679095853613056,
    "name": "Sea Turtle2",
    "type": "Sailboat",
    "length": 187,
    "owner": "117137237433967739406",
    "loads": [],
    "self": "https://assignment-9-magadia.uc.r.appspot.com/boats/5679095853613056"
}
```

Failure: Status 400

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

Failure: Status 401

Status 401 Unauthorized

Failure: Status 406

Status 406 Unacceptable

View all boats of a user

MIME Type of Response: application/json

Request	GET /boats
Method	GET
URL	`https://assignment-9-magadia.uc.r.appspot.com/boats`
Path Parameters	None
Request Body	None
Protected	Yes

Response Body Format: JSON, paginated with a “next” link and 5 results per page. Note: JWT authorization token must be given to use the “next” link to view the next page of results.

Outcome	Status Code	Notes
Success	200	
Failure	401	If JWT was missing or invalid

Example Responses:

Success

```
{
    "total_items": 10,
    "items": [
        {
            "id": "5146163359514624",
            "name": "The Pygmy Seahorse2",
            "type": "Sailboat",
            "length": 10,
            "owner": "111709994382238407003",
            "loads": [],
            "self": "https://assignment-9-
magadia.uc.r.appspot.com/boats/5146163359514624"
        },
        {
            "id": "5633378543992832",
            "name": "The Jellyfish",
            "type": "Submarine",
            "length": 45,
            "owner": "111709994382238407003",
            "loads": [],
            "self": "https://assignment-9-
magadia.uc.r.appspot.com/boats/5633378543992832"
        },
        {
            "id": "5637476211228672",
            "name": "The Jellyfish",
            "type": "Submarine",
            "length": 45,
            "owner": "111709994382238407003",
            "loads": [],
            "self": "https://assignment-9-
magadia.uc.r.appspot.com/boats/5637476211228672"
        },
        {
            "id": "5640825748848640",
            "name": "The Pygmy Seahorse",
            "type": "Sailboat",
            "length": 10,
            "owner": "111709994382238407003",
            "loads": [],
            "self": "https://assignment-9-
magadia.uc.r.appspot.com/boats/5640825748848640"
        },
        {
            "id": "5645760532054016",
            "name": "The Pygmy Seahorse3",
            "type": "Sailboat",
            "length": 10,
            "owner": "111709994382238407003",
            "loads": [],
            "self": "https://assignment-9-
magadia.uc.r.appspot.com/boats/5645760532054016"
```

```
        }
    ],
    "next": "https://assignment-9-
magadia.uc.r.appspot.com/boats/?cursor=CjESK2oWc35hc3NpZ25tZW50LTktbWFnYWRpYXIRCxIEQ
m9hdBiAgID4oNmDCgwYACAA"
}
```

Failure: Status 401
Status 401 Unauthorized

View a specific boat of a user

MIME Type of Response: application/json

Request	GET /boats/:boat_id
Method	GET
Example URL	`https://assignment-9-magadia.uc.r.appspot.com/boats/123123123`
Path Parameters	boat_id
Request Body	None
Protected	Yes

Response Body Format: JSON, paginated with a “next” link and 5 results per page. Note: JWT authorization token must be given to use the “next” link to view the next page of results.

Outcome	Status Code	Notes
Success	200	
Failure	401	If JWT was missing or invalid
Failure	403	If the JWT belonged to a valid user, but the boat did not belong to them
Failure	404	If the boat was not found
Failure	406	If the request does not accept JSON as a response

Example Responses:

Success: Status 200

```
{
    "id": "5679095853613056",
    "name": "Sea Turtle2",
    "type": "Sailboat",
    "length": 187,
    "owner": "117137237433967739406",
    "loads": [],
    "self": "https://assignment-9-magadia.uc.r.appspot.com/boats/5679095853613056"
}
```

Failure: Status 401
Status 401 Unauthorized

Failure: Status 403
{"Error": "The boat belongs to someone else"}

Failure: Status 404
{"Error": "No boat with that boat_id exists"}

Failure: Status 406
Status 401 Unacceptable

Delete a user's boat

Request	DELETE /boats/:boat_id
Method	DELETE
Example URL	`https://assignment-9-magadia.uc.r.appspot.com/boats/123123123`
Path Parameters	boat_id
Request Body	None
Protected	Yes
Response Body	None

Outcome	Status Code	Notes
Success	204	
Failure	404	This is returned if a boat with the given boat id was not found, and therefore could not be deleted.
	401	If JWT is missing or invalid
	403	If JWT belongs to a valid user but the boat belongs to another user

Example Responses:

Success: Status 204
Status 204 No Content

Failure: Status 401
Status 401 Unauthorized

Failure: Status 403
Status 403 Forbidden

Failure: Status 404
Status 404 Not Found

Update a user’s boat fully with PUT

MIME Types of Request Body Accepted: application/json
MIME Type of Response: application/json

Request	PUT /boats/:boat_id
Method	PUT
Example URL	`https://assignment-9-magadia.uc.r.appspot.com/boats/123123123`
Path Parameters	Boat_id
Request Body	Required

Request Body Format	JSON
Protected	Yes

Request JSON Attributes:

Name	Description	Required?
name	Name of boat	Yes
type	Type of boat (ex. Sailboat)	Yes
length	Length of boat in feet	Yes

Example Request Body

```
{
    "name": "The Walrus",
    "type": "yacht",
    "length": 1234
}
```

Response Body Format: JSON

Outcome	Status Code	Notes
Success	200	
Failure	400	This is returned if any of the 3 attributes are missing. Also, the values of any included attributes are not validated for data type. They are assumed to be valid.
Failure	401	If invalid or missing JWT is sent.
Failure	406	If the request does not accept application/json MIME type as a response.
Failure	404	If the boat was not found
	403	If the user provided a valid JWT but the boat belongs to another user

Example Responses:

Success: Status 200

```
{
    "id": "5712837116690432",
    "name": "The Walrus",
    "type": "yacht",
    "length": 1234,
    "loads": [],
    "self": "https://assignment-9-magadia.uc.r.appspot.com/boats/5712837116690432"
}
```

Failure: Status 404
{"Error": "No boat with this boat_id exists"}

Failure: Status 401
Status 401 Unauthorized

Failure: Status 406
Status 406 Unacceptable

Failure: Status 400
{"Error": "At least one of the required properties is missing"}

Failure: Status 403
{"Error": "The boat belongs to someone else"}

Update a user's boat partially with PATCH

MIME Types of Request Body Accepted: application/json
MIME Type of Response: application/json

Request	PATCH /boats/:boat_id
Method	PATCH
Example URL	`https://assignment-9-magadia.uc.r.appspot.com/boats/123123123`
Path Parameters	Boat_id
Request Body	Required
Request Body Format	JSON
Protected	Yes

Request JSON Attributes:

Name	Description	Required?
name	Name of boat	No
type	Type of boat (ex. Sailboat)	No
length	Length of boat in feet	No

Note: At least one valid attribute to be updated is required in the request body.

Example Request Body

```
{
    "name": "The Seahorse"
}
```

Response Body Format: JSON

Outcome	Status Code	Notes
Success	200	
Failure	400	This is returned if none of the attributes are provided for update.
Failure	401	If invalid or missing JWT is sent.
Failure	406	If the request does not accept application/json MIME type as a response.
Failure	404	If the boat was not found
	403	If the user provided a valid JWT but the boat belongs to another user

Example Responses:

Success: Status 200

```
{
    "id": "5712837116690432",
    "name": "The Seahorse",
    "type": "yacht",
    "length": 1234,
    "loads": [],
    "self": "https://assignment-9-magadia.uc.r.appspot.com/boats/5712837116690432"
}
```

Failure: Status 404

{"Error": "No boat with this boat_id exists"}

Failure: Status 401
Status 401 Unauthorized

Failure: Status 406
Status 406 Unacceptable

Failure: Status 400
{"Error": "No properties provided to update"}

Failure: Status 403
{"Error": "The boat belongs to someone else"}

Create a Load

MIME Types of Request Body Accepted: application/json
MIME Type of Response: application/json

Request	POST /loads
Method	POST
URL	`https://assignment-9-magadia.uc.r.appspot.com/loads`
Path Parameters	None
Request Body	Required
Request Body Format	JSON
Protected	No

Request JSON Attributes:

Name	Description	Required?
content	Contents of load (ex. Pencils)	Yes
quantity	The quantity of the content of the load	Yes

Example Request Body

```
{
    "content": "crayons",
    "quantity": 9999
}
```

Response Body Format: JSON

Outcome	Status Code	Notes
Success	201	
Failure	400	This is returned if either of the 2 attributes are missing. Also, the values of any included attributes are not validated for data type. They are assumed to be valid.
Failure	406	If the request does not accept JSON response

Example Responses:

Success Status 201

```
{
    "id": 5710353417633792,
    "content": "crayons",
    "quantity": 9999,
    "creation_date": "11/7/2021",
    "carrier": null,
    "self": "https://assignment-9-magadia.uc.r.appspot.com/loads/5710353417633792"
}
```

Failure: Status 400

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

Failure: Status 406

Status 406 Unacceptable

View all loads

MIME Type of Response: application/json

Request	GET /loads
Method	GET
URL	`https://assignment-9-magadia.uc.r.appspot.com/loads`
Path Parameters	None
Request Body	None
Protected	No

Response Body Format: JSON

Outcome	Status Code	Notes
Success	200	
Failure	406	If the request does not accept JSON response.

The successful response is paginated to 5 results per page.

Example Responses:

Success Status 200

```
{
    "total_items": 8,
    "items": [
        {
            "id": "5069549128908800",
            "quantity": 300,
            "carrier": null,
            "content": "crayons",
            "creation_date": "11/7/2021",
            "self": "https://assignment-9-
magadia.uc.r.appspot.com/loads/5069549128908800"
        },
        {
            "id": "5074526257807360",
            "quantity": 300,
            "carrier": null,
            "content": "crayons",
            "creation_date": "11/7/2021",
            "self": "https://assignment-9-
magadia.uc.r.appspot.com/loads/5074526257807360"
```

```
        },
        {
            "id": "5646488461901824",
            "quantity": 2,
            "carrier": null,
            "content": "erasers",
            "creation_date": "11/7/2021",
            "self": "https://assignment-9-magadia.uc.r.appspot.com/loads/5646488461901824"
        },
        {
            "id": "5666707490209792",
            "quantity": 300,
            "carrier": null,
            "content": "crayons",
            "creation_date": "11/7/2021",
            "self": "https://assignment-9-magadia.uc.r.appspot.com/loads/5666707490209792"
        },
        {
            "id": "5667525748588544",
            "quantity": 100,
            "carrier": null,
            "content": "pencils",
            "creation_date": "11/7/2021",
            "self": "https://assignment-9-magadia.uc.r.appspot.com/loads/5667525748588544"
        }
    ],
    "next": "https://assignment-9-magadia.uc.r.appspot.com/loads/?cursor=CjESK2oWc35hc3NpZ25tZW50LTktbWFnYWRpYXIRCxIETG9hZBiAgIDY2tKICgwYACAA"
}
```

Failure: Status 406
Status 406 Unacceptable

View a specific load

MIME Type of Response: application/json

Request	GET /loads/:load_id
Method	GET
Example URL	`https://assignment-9-magadia.uc.r.appspot.com/loads/131923`
Path Parameters	Load_id
Request Body	None
Protected	No

Response Body Format: JSON

Outcome	Status Code	Notes
Success	200	
Failure	404	If the load was not found
Failure	406	If the request does not accept a JSON response

Example Responses:

Success: Status 200

```
{
    "id": 5710353417633792,
    "content": "crayons",
    "quantity": 9999,
    "carrier": null,
    "creation_date": "11/7/2021",
    "self": "https://assignment-9-magadia.uc.r.appspot.com/loads/5710353417633792"
}
```

Failure: Status 406
Status 406 Unacceptable

Success: Status 404
{"Error": "No load with this load_id exists"}

Delete a load

Request	DELETE /loads/:load_id
Method	DELETE
Example URL	`https://assignment-9-magadia.uc.r.appspot.com/loads/123123123`

Path Parameters	load_id
Request Body	None
Protected	No
Response Body	None

Outcome	Status Code	Notes
Success	204	
Failure	404	This is returned if a load with the given load id was not found, and therefore could not be deleted.

Example Responses:

Success: Status 204
Status 204 No Content

Failure: Status 404
Status 404 Not Found

Update a load fully with PUT

MIME Types of Request Body Accepted: application/json
MIME Type of Response: application/json

Request	PUT /loads/:load_id
Method	PUT
Example URL	`https://assignment-9-magadia.uc.r.appspot.com/loads/123123123`
Path Parameters	load_id
Request Body	Required
Request Body Format	JSON
Protected	No

Request JSON Attributes:

Name	Description	Required?
content	Contents of load (ex. Pencils)	Yes
quantity	The quantity of the content of the load	Yes

Example Request Body

```
{
    "content": "colored pencils",
    "quantity": 200
}
```

Response Body Format: JSON

Outcome	Status Code	Notes
Success	200	
Failure	400	This is returned if any of the 2 attributes are missing. Also, the values of any included attributes are not validated for data type. They are assumed to be valid.
Failure	406	If the request does not accept application/json MIME type as a response.
Failure	404	If the load was not found

Example Responses:

Success: Status 200

```
{
    "id": "5710353417633792",
    "content": "colored pencils",
    "quantity": 200,
    "carrier": null,
    "creation_date": "11/7/2021",
    "self": "https://assignment-9-magadia.uc.r.appspot.com/loads/5710353417633792"
}
```

Failure: Status 404

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

Failure: Status 406
Status 406 Unacceptable

Failure: Status 400
{"Error": "All properties have to be updated for PUT"}

Update a load partially with PATCH

MIME Types of Request Body Accepted: application/json
MIME Type of Response: application/json

Request	PATCH /loads/:load_id
Method	PATCH
Example URL	`https://assignment-9-magadia.uc.r.appspot.com/loads/123123123`
Path Parameters	load_id
Request Body	Required
Request Body Format	JSON
Protected	No

Request JSON Attributes:

Name	Description	Required?
content	Contents of load (ex. Pencils)	No
quantity	The quantity of the content of the load	No

Note: At least one valid attribute to be updated is required in the request body.

Example Request Body

```
{
    "content": "markers"
}
```

Response Body Format: JSON

Outcome	Status Code	Notes
Success	200	
Failure	400	This is returned if neither of the attributes are given for update in the request body
Failure	406	If the request does not accept application/json MIME type as a response.
Failure	404	If the load was not found

Example Responses:

Success: Status 200

```
{
    "id": "5710353417633792",
    "content": "markers",
    "quantity": 200,
    "creation_date": "11/7/2021",
    "carrier": null,
    "self": "https://assignment-9-magadia.uc.r.appspot.com/loads/5710353417633792"
}
```

Failure: Status 404

{"Error": "No boat with this load_id exists"}

Failure: Status 406

Status 406 Unacceptable

Failure: Status 400

{"Error": "No properties provided to update"}

Managing Loads on Boats:

1. Assign a Load to a Boat

Request	PUT /boats/:boat_id/loads/:load_id
Path Parameters	id of boat, id of load
Request Body	None
Protected	No

Response Body Format: JSON

Outcome	Status Code	Notes
Success	204	
Failure	404	This is returned if no load with the given load_id is found and/or if no boat with the given boat_id is found.
Failure	403	This is returned if the specified load is currently assigned to another boat.

Example Responses:

Success
Status: 204 No Content

Failure: Status 404

```
{"Error": "The specified boat and/or load does not exist"}
```

Failure: Status 403

```
{"Error": "The specified load is assigned to another boat"}
```

2. Remove a Load from a Boat

Request	DELETE /boats/:boat_id/loads/:load_id

Path Parameters	id of boat, id of load
Request Body	None
Protected	No

Response Body Format: JSON

Outcome	Status Code	Notes
Success	204	
Failure	404	This is returned if no load with the given load_id is found and/or if no boat with the given boat_id is found.
Failure	403	This is returned if the specified load is not currently assigned to the specified boat, and therefore it cannot be removed.

Example Responses:

Success
Status: 204 No Content

Failure: Status 404
`{"Error": "The specified boat and/or load does not exist"}`

Failure: Status 403
`{"Error": "The specified load is not assigned to the specified boat"}`

View all users

MIME Type of Response: application/json

Request	GET /users
Method	GET
URL	`https://assignment-9-magadia.uc.r.appspot.com/users`
Path Parameters	None
Request Body	None

Protected	No

Response Body Format: JSON

Outcome	Status Code	Notes
Success	200	
Failure	406	If the request does not accept JSON response.

Example Responses:

Success Status 200

```
{
    "items": [
        {
            "user_id": "117137237433967739406"
        },
        {
            "user_id": "111709994382238407003"
        }
    ],
    "total_items": 2
}
```

Failure: Status 406

Status 406 Unacceptable

Invalid Endpoints:

Request	DELETE /boats
Method	DELETE
URL	`https://assignment-9-magadia.uc.r.appspot.com/boats`
Path Parameters	None
Request Body	None
Protected	No
Response Body	None

Request	PUT /boats
Method	PUT
URL	`https://assignment-9-magadia.uc.r.appspot.com/boats`
Path Parameters	None
Request Body	None
Protected	No
Response Body	None

Request	PATCH /boats
Method	PATCH
URL	`https://assignment-9-magadia.uc.r.appspot.com/boats`
Path Parameters	None
Request Body	None
Protected	No
Response Body	None

Request	DELETE /loads
Method	DELETE
URL	`https://assignment-9-magadia.uc.r.appspot.com/loads`
Path Parameters	None
Request Body	None
Protected	No
Response Body	None

Request	PUT /loads
Method	PUT
URL	`https://assignment-9-magadia.uc.r.appspot.com/loads`
Path Parameters	None
Request Body	None
Protected	No
Response Body	None

Request	PATCH /loads
Method	PATCH
URL	`https://assignment-9-magadia.uc.r.appspot.com/loads`
Path Parameters	None
Request Body	None
Protected	No
Response Body	None

Outcome	Status Code	Notes
Failure	405	These are invalid endpoints

Example Response:

Failure: Status 405
Status 405 Method Not Allowed