Portfolio Project: API Spec

CS 493: Cloud Application Development

Spring 2022

Oregon State University

Last Update: June 5th, 2022

Change log

Version	Change	Date
1.0	Initial version.	June 5, 2022

App URL

https://portfolio-glowackp.uc.r.appspot.com (login at this URL as well – using "Sign In" button)

Data Model

The app stores three kinds of entities in Datastore: Users, Boats and Loads.

Users

- Related to Boats

Property	Data Type	Notes
id	Integer	The id of the user. Datastore automatically generates it.
name	String	Email associated with the user (i.e. name property of JWT).
user_id	String	**Unique** user id for the user (i.e. sub property of the JWT token). Each boat has an *owner* attribute that matches the User’s *user_id*.

Boats

- Related to both Users and Loads

Property	Data Type	Notes
id	Integer	The id of the boat. Datastore automatically generates it. Don't add it yourself as a property of the entity.
owner	String	Unique id of the owner – matched to the *user_id* of the *Users* Data Model (see above).
name	String	Name of the boat.
type	String	Type of the boat. E.g., Sailboat, Catamaran, etc.
length	Integer	The length of the boat in feet.
loads	Array of Integers	Loads contained on the boat. Can be either null an array of objects.
loads.id	Integer	See *Loads Model* below.

Loads

- Related to Boats

Property	Data Type	Notes
id	Integer	The id of the load. Datastore automatically generates it. Don't add it yourself as a property of the entity.
volume	Integer	The volume of the load.
carrier	Object	The boat carrying the load. Can be either null or a single object.
carrier.id	Integer	See *Boats Model* above.
carrier.name	String	See *Boats Model* above.
item	String	Description of the load.
creation_date	String	Date load was created in format MM/DD/YYYY

List all Users

List all the users.

```
GET /users
```

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	The client sends a request with Accept header set to a MIME type that is not supported.

Response Examples

Success

```
Status: 200 OK

[
    {
        "name": "wallace@cheese.com",
        "user_id": "auth0|62923b18f9d32a006f916599",
        "id": 5634161670881280
    },
    {
        "user_id": "auth0|629a36f28ae5ce006907345c",
        "name": "jose.altuve@astros.com",
        "id": 5644004762845184
    }
]
```

Failure

```
Status: 406 Not Acceptable
{
    "Error": "Server only sends application/json data."
}
```

Create a Boat

Allows you to create a new boat.

```
POST /boats
```

Request

Authorization

Requires a valid JWT token (bearer token)

Path Parameters

None

Request Body

Required

Request MIME Type

Success: application/json **only**

Request JSON Attributes

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

Request Input Validation (Error = Status Code 400)

- All three (3) attributes (name, type, length) are required.
- The attributes must have the following types:
  - Name: string
  - Type: string
  - Length: number
- The request body must not have any other attributes besides the three listed above.
- The name attribute must be a string of 50 or less characters.

Request Body Example

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

Response

Response MIME Type

Success: application/json **only**

Failure**:** application/json

Response Statuses

Outcome	Status Code	Notes

Success	201 Created	
Failure	400 Bad Request	If the request body fails any of the input validation checks lists in *Request Input Validation*).
Failure	401 Unauthorized	If the request is not sent with a valid JWT token (Bearer token)
Failure	406 Not Acceptable	The client sends a request with Accept header set to a MIME type that is not supported.
Failure	415 Unsupported Media Type	The client sends an unsupported MIME type (i.e. Content-Type in header)

Response Examples

Success

```
Status: 201 Created
{
    "length": 28,
    "name": "Sea Witch",
    "type": "Catamaran",
    "id": 5710353417633792,
    "self": "http://localhost:8080/boats/5710353417633792"
}
```

Failure

```
Status: 400 Bad Request
{
    "Error": "Request has invalid data."
}

Status: 401 Unauthorized
{
    "Error": "Invalid token"
}

Status: 406 Not Acceptable
{
    "Error": "Server only sends application/json data."
}

Status: 415 Unsupported Media Type
{
    "Error": "Server only accepts application/json data."
}
```

Get a Boat

Allows you to get an existing boat

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

Request

Authorization

Requires a valid JWT token (bearer token) associated with the owner for the boat_id.

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If the request is not sent with a valid JWT token (Bearer token)
Failure	403 Forbidden	A valid JWT was provided but not the correct token that corresponds to the owner of the boat.
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	The client sends a request with Accept header set to a MIME type that is not supported.

Response Examples

Success

```
Status: 200 OK

{
    "length": 28,
    "name": "Sea Witch",
    "loads": [],
    "type": "Catamaran",
    "id": 5631671361601536,
    "self": "http://localhost:8080/boats/5631671361601536"
}
```

Failure

```
Status: 401 Unauthorized

{
    "Error": "Invalid token"
}
```

Status: 403 Forbidden

```
{
    "Error": "Forbidden"
}
```

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{
    "Error": "Server only sends application/json data."
}
```

List all Boats for an Owner

List all the boats.

```
GET /boats
```

Request

Authorization

Requires a valid JWT token (bearer token)

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If the request is not sent with a valid JWT token (Bearer token)
Failure	406 Not Acceptable	The client sends a request with Accept header set to a MIME type that is not supported.

Response Examples

- Implements pagination with limit of 5 boats per page
- There is a next link to get the next page of results, unless there are no more pages of result
- The total number of items in the collection is provided as “*total_items*”

Success

Status: 200 OK

```
{
    "boats": [
        {
            "length": 13,
            "name": "Calypso",
            "loads": [],
            "owner": "auth0|62923b18f9d32a006f916599",
            "type": "Sailing Boat",
            "id": 5631671361601536,
            "self": "http://localhost:8080/boats/5631671361601536"
        },
        {
            "loads": [],
            "length": 200,
            "owner": "auth0|62923b18f9d32a006f916599",
            "type": "Yacht",
            "name": "Stonks",
            "id": 5635008819625984,
            "self": "http://localhost:8080/boats/5635008819625984"
        },
        {
            "loads": [],
            "owner": "auth0|62923b18f9d32a006f916599",
            "type": "Yacht",
            "length": 18,
            "name": "401K",
            "id": 5638358357245952,
            "self": "http://localhost:8080/boats/5638358357245952"
        },
        {
            "type": "Kayak",
            "length": 8,
            "owner": "auth0|62923b18f9d32a006f916599",
            "loads": [],
            "name": "Lesiure",
            "id": 5643280054222848,
            "self": "http://localhost:8080/boats/5643280054222848"
        },
        {
            "owner": "auth0|62923b18f9d32a006f916599",
            "loads": [],
            "type": "Catamaran",
            "length": 28,
            "name": "Sea Witch",
            "id": 5646488461901824,
            "self": "http://localhost:8080/boats/5646488461901824"
        }
    ],
    "next": "http://localhost:8080/boats?cursor=Ci8SKWoUc35wb3J0Zm9saW8tZ2xvd2Fja3By
EQsSBEJvYXQYgICA2LjugwoMGAAgAA==",
    "total_items": 7
}
```

Failure

Status: 401 Unauthorized

```
{
    "Error": "Invalid token"
}
```

Status: 406 Not Acceptable

```
{
    "Error": "Server only sends application/json data."
}
```

Delete a Boat

Allows you to delete a boat. Note that deleting a boat will unload any loads that were loaded on to it.

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

Request

Authorization

Requires a valid JWT token (bearer token) associated with the owner for the boat_id.

Path Parameters

Name	**Description**
boat_id	ID of the boat

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	If the request is not sent with a valid JWT token (Bearer token)
Failure	403 Forbidden	A valid JWT was provided but not the correct token that corresponds to the owner of the boat.
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized

{
    "Error": "Invalid token"
}

Status: 403 Forbidden

{
    "Error": "Forbidden"
}
```

Status: 404 Not Found

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

Edit a Boat (All Attributes)

Allows you to update a boat with all attributes provided in the request. Given this is a replace, the *loads* attribute is reset to an empty array and the associated relationships are removed.

PUT /boats/:boat_id

Request

Authorization

Requires a valid JWT token (bearer token) associated with the owner for the boat_id.

Path Parameters

Name	**Description**
boat_id	ID of the boat

Request Body

Required

Request MIME Type

Success: application/json **only**

Request JSON Attributes

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

Request Input Validation (Error = Status Code 400)

- All three (3) attributes (name, type, length) are required.
- The attributes must have the following types:
  - Name: string
  - Type: string
  - Length: number
- The request body must not have any other attributes besides the three listed above.
- The name attribute must be a string of 50 or less characters.

Request Body Example

```
{
    "name": "Lucia",
    "type": "Catamaran",
    "length": 20
}
```

Response

Response MIME Type

Success: None

Failure: application/json

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	If the request body fails any of the input validation checks lists in *Request Input Validation*).
Failure	401 Unauthorized	If the request is not sent with a valid JWT token (Bearer token)
Failure	403 Forbidden	A valid JWT was provided but not the correct token that corresponds to the owner of the boat.
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	The client sends a request with Accept header set to a MIME type that is not supported.
Failure	415 Unsupported Media Type	The client sends an unsupported MIME type (i.e. Content-Type in header)

Response Examples

Failure

```
Status: 400 Bad Request
{
    "Error": "Request has invalid data."
}

Status: 403 Forbidden
{
    "Error": "Name violates uniqueness constraint."
}

Status: 404 Not Found

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

Status: 406 Not Acceptable
{
    "Error": "Server only sends application/json data."
}

Status: 415 Unsupported Media Type
{
    "Error": "Server only accepts application/json data."
}
```

Edit a Boat (Some Attributes)

Allows you to update a boat with one or more of the required attributes, besides the *loads* attribute. The *loads* and the associated relationships remain unchanged.

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

Request

Authorization

Requires a valid JWT token (bearer token) associated with the owner for the boat_id.

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request MIME Type

Success: application/json **only**

Request JSON Attributes

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

Request Input Validation (Error = Status Code 400)

- **One** of three (3) attributes (name, type, length) is required.
- The attributes must have the following types:
  - Name: string
  - Type: string
  - Length: number
- The request body must not have any other attributes besides the three listed above.
- The name attribute must be a string of 50 or less characters.

Request Body Example

```
{
    "name": "The Patrick"
}
```

Response

Response MIME Type

Success: None

Failure: application/json

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	If the request body fails any of the input validation checks lists in *Request Input Validation*).
Failure	401 Unauthorized	If the request is not sent with a valid JWT token (Bearer token)
Failure	403 Forbidden	A valid JWT was provided but not the correct token that corresponds to the owner of the boat.
Failure	404 Not Found	No boat with this boat_id exists
Failure	415 Unsupported Media Type	The client sends an unsupported MIME type (i.e. Content-Type in header)

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 400 Bad Request
{
    "Error": "Request has invalid data."
}

Status: 403 Forbidden
{
    "Error": "Name violates uniqueness constraint."
}

Status: 404 Not Found

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

Status: 415 Unsupported Media Type
{
    "Error": "Server only accepts application/json data."
}
```

Create a Load

Allows you to create a new load. All newly created loads should begin unassigned to any boat.

```
POST /loads
```

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	Yes
item	Description of the load.	Yes
creation_date	Date load was created.	Yes

Request Body Example

```
{
  "volume": 5,
  "item": "LEGO Blocks",
  "creation_date": "10/18/2021"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request body is missing any of the required attributes.
Failure	406 Not Acceptable	The client sends a request with Accept header set to a MIME type that is not supported.

Response Examples

Success

```
Status: 201 Created
{
    "id": 5068721408180224,
    "volume": 5,
    "item": "LEGO Blocks",
    "carrier": null,
    "creation_date": "10/18/2021",
    "self": "http://localhost:8080/loads/5068721408180224"
}
```

Failure

Status: 400 Bad Request

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

Status: 406 Not Acceptable

```
{
    "Error": "Server only sends application/json data."
}
```

Get a Load

Allows you to get an existing load using its load_id.

```
GET /loads/load_id
```

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No load with load_id exists.
Failure	406 Not Acceptable	The client sends a request with Accept header set to a MIME type that is not supported.

Response Examples

Success

```
Status: 200 OK

{
    "creation_date": "10/18/2021",
    "carrier": null,
    "item": "LEGO Blocks",
    "volume": 5,
    "id": 5068721408180224,
    "self": "http://localhost:8080/loads/5068721408180224"
}
```

Failure

```
Status: 404 Not Found

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

Status: 406 Not Acceptable

{
    "Error": "Server only sends application/json data."
}
```

List all Loads

List all the loads.

```
GET /loads
```

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	The client sends a request with Accept header set to a MIME type that is not supported.

Response Examples

- Implements pagination with limit of 5 loads per page
- There is a next link to get the next page of results, unless there are no more pages of result
- The total number of items in the collection is provided as “*total_items*”

Success

```
Status: 200 OK

{
    "loads": [
        {
            "item": "Stuffed Animals",
            "creation_date": "10/19/2021",
            "carrier": null,
            "volume": 10,
            "id": 5633378543992832,
            "self": "http://localhost:8080/loads/5633378543992832"
        },
        {
            "item": "Books",
            "volume": 11,
            "creation_date": "10/21/2021",
            "carrier": null,
            "id": 5637476211228672,
            "self": "http://localhost:8080/loads/5637476211228672"
        },
        {
            "carrier": null,
            "creation_date": "10/21/2021",
            "item": "Toys",
            "volume": 4,
            "id": 5640825748848640,
            "self": "http://localhost:8080/loads/5640825748848640"
        },
        {
            "creation_date": "10/18/2021",
            "volume": 5,
            "carrier": null,
            "item": "LEGO Blocks",
            "id": 5673082664517632,
            "self": "http://localhost:8080/loads/5673082664517632"
        },
        {
            "item": "Food",
            "creation_date": "10/19/2021",
            "carrier": null,
            "volume": 3,
            "id": 5705808872472576,
            "self": "http://localhost:8080/loads/5705808872472576"
        }
    ],
    "next": "http://localhost:8080/loads?cursor=Ci8SKWoUc35wb3J0Zm9saW8tZ2xvd2Fja3By
EQsSBExvYWQYgICAuPKskQoMGAAgAA==",
    "total_items": 6
}
```

Failure

Status: 406 Not Acceptable

```
{
    "Error": "Server only sends application/json data."
}
```

Edit a Load (All Attributes)

Allows you to update a load with all attributes provided in the request. Given this is a replace, the *carrier* attribute is reset to an empty array and the associated relationships are removed.

PUT /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request MIME Type

Success: application/json **only**

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	Yes
item	Description of the load.	Yes
creation_date	Date load was created.	Yes

Request Body Example

```
{
  "volume": 25,
  "item": "LEGO Blocks",
  "creation_date": "06/05/2022"
}
```

Response

Response MIME Type

Success: None

Failure: application/json

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	If the request body is missing any of the required attributes.
Failure	404 Not Found	No load with this load_id exists

Response Examples

Failure

```
Status: 400 Bad Request
{
    "Error": "Request has invalid data."
}

Status: 404 Not Found

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

Edit a Load (Some Attributes)

Allows you to update a load with one or more attributes provided in the request, except the *carrier* attribute. The carrier and its associated relationships remain unchanged.

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

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request MIME Type

Success: application/json **only**

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	No
item	Description of the load.	No
creation_date	Date load was created.	No

Request Body Example

```
{
  "volume": 50
}
```

Response

Response MIME Type

Success: None

Failure: application/json

Response Statuses

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

Response Examples

Failure

```
Status: 404 Not Found

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

Delete a Load

Allows you to delete a load. Deleting a load should update the boat that was carrying it.

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

Request

Path Parameters

Name	Description
load_id	ID of the load

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"
}
```

Assign a Load to Boat

Allows you to assign a load to a specific boat, given that boat is owned by the user providing the JWT. This will update both the load and boat involved in the transaction.

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

Request

Authorization

Requires a valid JWT token (bearer token) associated with the owner for the boat_id.

Path Parameters

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

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 currently unassigned.
Failure	400 Bad Request	The load is already assigned to a boat.
Failure	401 Unauthorized	If the request is not sent with a valid JWT token (Bearer token)
Failure	403 Forbidden	A valid JWT was provided but not the correct token that corresponds to the owner of the boat.
Failure	404 Not Found	The specified boat and/or load does not exist

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 400 Bad Request

{
    "Error": "The load is already loaded on another boat"
}
```

Status: 401 Unauthorized

```
{
    "Error": "Invalid token"
}
```

Status: 403 Forbidden

```
{
    "Error": "Forbidden"
}
```

Status: 404 Not Found

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

Comment

- A boat can be assigned multiple loads.
- A load can only be assigned to one boat.

Remove a Load from Boat

Allows you to remove an assignment of a load on a specific boat, given that boat is owned by the user providing the JWT. This will update both the load and boat involved in the transaction.

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

Request

Authorization

Requires a valid JWT token (bearer token) associated with the owner for the boat_id.

Path Parameters

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

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

- Removing a load from a boat updates the carrier attribute of the load data model, as well as the loads attribute of the boat data model.

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 currently assigned to the specified boat.
Failure	401 Unauthorized	If the request is not sent with a valid JWT token (Bearer token)
Failure	403 Forbidden	A valid JWT was provided but not the correct token that corresponds to the owner of the boat.
Failure	404 Not Found	The specified boat and/or load does not exist, or the load is not currently assigned to the specified boat.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 400 Bad Request

{
    "Error": " No boat with this boat_id is loaded with the load with this load_id"
}
```

Status: 401 Unauthorized

```
{
    "Error": "Invalid token"
}
```

Status: 403 Forbidden

```
{
    "Error": "Forbidden"
}
```

Status: 404 Not Found

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