

Portfolio Assignment: API Spec

CS 493: Cloud Application Development

Fall 2020

Oregon State University

Last Update Dec 2, 2020. 2:30 pm Pacific

Application deployed at: <https://hartwelg-final-gae.wl.r.appspot.com>

Login at: <https://hartwelg-final-gae.wl.r.appspot.com>

Change log	2
Create a Boat	2
Get a Boat	6
List all Boats	7
Edit a Boat.....	8
Delete a Boat	11
Create a Load	12
Get a Load	15
List all Loads.....	17
Delete a Load	19
Create an Owner.....	20
Get an Owner.....	22
List all Owners.....	24
Delete an owner	26
Load put on Boat	27
Load removed from Boat.....	29
Boat assigned to Owner.....	30
Boat removed from Owner.....	32
View Loads for a Boat	33
View Boats for an Owner.....	35

Change log

Page	Change	Date
	Initial version.	M, Oct 12
	Final Project version.	W, Dec 2

Create a Boat

Allows you to create a new boat.

POST /boats

Request

Request 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
Owner	String	User_id of the owner of the boat	Yes
Loads	List	list of the loads on the boat	Yes

Request Body Example

```
{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": null,
  "loads": []
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	<p>If the request is missing any of the 5 required attributes, the boat will not be created, and 400 status code will be returned.</p> <p>If the value of the boat's "length" attribute exceeds 500, the boat will not be created and 400 status code will be returned.</p> <p>If the boat's name exceeds 15 characters or the name contains non-alphanumeric characters, the boat will not be created, and 400 status code will be returned.</p>
Failure	401 Unauthorized	If the user's access token is non-existent or the token does not verify, the boat will not be created, and 401 status code will be returned.
failure	406 Not Acceptable	If the request to the endpoint does not have 'application/json' in the "Accept" header, the boat will not be created, and 406 status code will be returned.
Failure	415 Unsupported Media Type	If the request to the endpoint does not have "application/json" in the "Content-Type" header, the boat will not be created and 415 status code will be returned.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. This value needs to be sent in the response body as shown in the example.
- 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. You must not store the `self` attribute in Datastore

Success

Status: 201 Created

```
{
  "id": "abc123",
  "name": "Sea Witch",
```

```
"type": "Catamaran",  
"length": 28,  
"self": "https://<your-app>/boats/abc123"  
}
```

Failure

Status: 400 Bad Request

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

OR

```
{  
"Error": "Boat name is too long or contains non-alphanumeric characters"  
}
```

OR

```
{  
"Error": "Boat length cannot exceed 500"  
}
```

OR

```
{  
"Error": "Extraneous field(s) in request"  
}
```

Failure

Status: 401 Unauthorized

```
{  
"Error": "The request does not have proper authorization"  
}
```

Failure

Status: 403 Forbidden

```
{  
"Error": "Boat name is not unique"  
}
```

Failure

Status: 406 Not Acceptable

```
{
```

```
}
```

Failure

Status: 415 Unsupported Media Type

```
{
```

```
}
```

Get a Boat

Allows you to get an existing boat

GET /boats/:boat_id

Request

Request Parameters

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

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No boat with this boat_id exists
Failure	401 Unauthorized	User is not authorized to view this boat

Response Examples

Success

Status: 200 OK
<pre>{ "id": "abc123", "name": "Sea Witch", "type": "Catamaran", "length": 28, "owner": null, "self": "https://<your-app>/boats/abc123" }</pre>

Failure

Status: 404 Not Found
<pre>{ "Error": "No boat with this boat_id exists" }</pre>

Failure

Status: 401 Unauthorized

```
{
  "Error": "The request does not have proper authorization"
}
```

List all Boats

List all the boats.

GET /boats

Request

Request Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK

```
[
  {
    "id": "abc123",
    "name": "Sea Witch",
    "type": "Catamaran",
    "length": 28,
    "owner": null,
    "self": "https://<your-app>/boats/abc123"
  },
  {
    "id": "def456",
    "name": "Adventure",
```

```

    "type": "Sailboat",
    "length": 50,
    "owner": null,
    "self": "https://<your-app>/boats/def456"
  },
  {
    "id": "xyz123",
    "name": "Hocus Pocus",
    "type": "Sailboat",
    "length": 100,
    "owner": null,
    "self": "https://<your-app>/boats/xyz123"
  }
]

```

Edit a Boat

Allows you to edit a boat. Protected endpoint.

PUT /boats/:boat_id

Request

Request Parameters

Name	Type	Description	Required?
boat_id	String	ID of the boat	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
Owner	String	User_id of the owner of the boat	Yes
Loads	List	List of loads on the boat	No

Request Body Example

```

{
  "name": "Sea Witch",
  "type": "Catamaran",

```



```
"length": 99,  
"owner": null,  
"loads": []  
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	If the request is not authorized, the boat will not be updated and 401 status code will be returned.
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

Success

```
Status: 200 OK  
  
{  
  
}
```

Failure

```
Status: 401 Unauthorized  
  
{  
"Error": "The request does not have proper authorization"  
}
```

```
Status: 404 Not Found
```

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

Delete a Boat

Allows you to delete a boat. If the boat has any loads or is owned by an owner, the loads will be unloaded and the boat will be removed from the owner. Protected endpoint.

DELETE /boats/:boat_id

Request

Request Parameters

Name	Type	Description	Required?
boat_id	String	ID of the boat	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 boat with this boat_id exists
Failure	401 Unauthorized	Request is not authorized

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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

Failure

Status: 401 Unauthorized

```
{
  "Error": "The request does not have proper authorization"
}
```

Create a Load

Allows you to create a new load. Unprotected endpoint.

POST /loads

Request

Request Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
Weight	Integer	The weight of the load	Yes
Content	String	The content of the load	Yes
Delivery_Date	String	The delivery date of the load	Yes
Carrier	Integer	The id of the boat carrying this load	No

Request Body Example

```
{
  "Weight": 20,
  "Content": "Legos",
  "Delivery_date": "10/10/2010",
  "carrier": null,
  "id": "123456789",
  "self": "http://<app_url>/loads/<load_id>"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	<p>If the request is missing the number attribute, the slip must not be created, and 400 status code must be returned.</p> <p>You don't need to validate the values of this attribute and can assume that if the number attribute is specified, then its value is valid.</p> <p>You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute other than number).</p>

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. This value needs to be sent in the response body as shown in the example.
- The value of the attribute `carrier` is the ID of the boat currently carrying this load. If there is no boat carrying this load, the value of `carrier` should be null
- The value of the attribute `self` is a live link to the REST resource corresponding to this slip. In other words, this is the URL to get this newly created slip. You must not store this attribute in Datastore

Success

Status: 201 Created
<pre>{ "id": "123abc", "Weight": 20, "Content": "Legos",</pre>

```
{
  "Delivery_date": "10/10/2010",
  "Carrier": null,
  "self": "https://<your-app>/slips/123abc"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object is missing the required number"
}
```

Get a Load

Allows you to get an existing load. Unprotected endpoint.

```
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

Response Examples

Success

Status: 200 OK

```
{
  "id": "123abc",
  "weight": 20,
  "content": "Legos",
  "delivery_date": "10/10/2010",
  "carrier": "abc123",
  "self": "https://<your-app>/slips/123abc"
}
```

Failure

Status: 404 Not Found

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


List all Loads

List all the loads. Unprotected endpoint.

GET /loads

Request

Request Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK

```
[
{
  "id": "123abc",
  "weight": 20,
  "content": "Legos",
  "delivery_date": "10/10/2010",
  "carrier": "abc123",
  "self": "https://<your-app>/slips/123abc"
},
{
  "id": "456def",
```

```
"weight": 200,  
"content": "Hover bike",  
"Delivery_date": "10/25/2015",  
"carrier": null,  
"self": "https://<your-app>/slips/456def"  
}  
]
```

Delete a Load

Allows you to delete a load. If the load being deleted is on a boat, the load is also “unloaded” from the boat. Unprotected endpoint.

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"
}
```

Create an Owner

Allows you to create a new owner. Protected endpoint.

POST /owners

Request

Request Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
password	String	The owner's password	Yes
Age	Int	The owner's age	Yes
boats	List	The boats owned by this owner	Yes

Request Body Example

```
{  
  "pass": "password",  
  "age": 40,  
  "boats": []  
}
```

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 required attributes, the owner will not be put into the system and 400 status code will be returned.
Failure	401 Unauthorized	If the user does not have proper authorization, the owner entity will not be created and 401 status code will be returned

Response Examples

Success

Status: 201 Created

```
{
  "user_id": "105294518571022308998",
  "password": "password",
  "age": 40,
  "boats": [],
  "id": "6236692331626496",
  "self": "http://localhost:8080/owners//6236692331626496"
}
```

Failure

Status: 400 Bad Request

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

Failure

Status: 401 Unauthorized

```
{
  "Error": "The request does not have proper authorization"
}
```

```
}
```

Get an Owner

Allows you to get an existing owner. Unprotected endpoint.

```
GET /owners/:owner_id
```

Request

Request Parameters

Name	Type	Description	Required?
owner_id	String	ID of the owner	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

Response Examples

Success

Status: 200 OK

```
{
  "self": "http://localhost:8080/owners/5163227868561408",
  "boats": [
    {
      "owner": "105294518571022308998",
      "id": "4817117325033472",
```

```
      "self": "http://localhost:8080/boats/4817117325033472"
    }
  ],
  "id": 5163227868561408,
  "user_id": "105294518571022308998"
}
```

Failure

Status: 404 Not Found

```
{
  "Error": "No owner with this owner_id exists"
}
```

List all Owners

List all the owners. Unprotected endpoint.

GET /owners

Request

Request Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK

```
{
  "owners": [
    {
      "self": "http://localhost:8080/owners/5163227868561408",
      "boats": [
        {
          "owner": "105294518571022308998",
          "id": "4817117325033472",
          "self": "http://localhost:8080/boats/4817117325033472"
        }
      ]
    }
  ]
}
```



```
    ],  
    "id": 5163227868561408,  
    "user_id": "105294518571022308998"  
  }  
{  
  "self": "http://localhost:8080/owners/5163227868561408",  
  "boats": [  
    {  
      "owner": "105294518571022308998",  
      "id": "4817117325035631",  
      "self": "http://localhost:8080/boats/4817117325035631"  
    }  
  ],  
  "id": 5163227868561408,  
  "user_id": "105294518571022308998"  
}  
]  
}
```

Delete an owner

Allows you to delete an owner. Protected endpoint.

DELETE /owners/:owner_id

Request

Request Parameters

Name	Type	Description	Required?
owner_id	String	ID of the owner	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 owner with this owner_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
  "Error": "No owner with this owner_id exists"
}
```

Load put on Boat

A load is put on a boat. Unprotected endpoint.

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	201 Created	Succeeds only if a boat exists with this boat_id, a load exists with this load_id and the load is not already on the boat.
Failure	403 Forbidden	There is already a load on this boat.
Failure	403 Forbidden	The load is already on another boat
Failure	404 Not Found	No boat with this boat_id exists, and/or no load with this load_id exists.

Response Examples

Success

Status: 201 Created

Failure

Status: 403 Forbidden

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

Status: 404 Not Found

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

Comment

- A load cannot be assigned to multiple boats.

Load removed from Boat

Load has been removed from the boat. Unprotected endpoint.

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	404 Not Found	No load with this load_id is on the boat with this boat_id.

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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

Boat assigned to Owner

A boat is assigned to an owner. Unprotected endpoint.

PUT /owners/:owner_id/boats/:boat_id

Request

Request Parameters

Name	Type	Description	Required?
owner_id	String	ID of the owner	Yes
boat_id	String	ID of the boat	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	201 Created	Succeeds only if an owner exists with this owner_id, a boat exists with this boat_id and the boat is not already owned by someone.
Failure	403 Forbidden	There is already a boat owned by this owner.
Failure	403 Forbidden	The boat is owned by another owner.
Failure	404 Not Found	No boat with this boat_id exists, and/or no owner with this owner_id exists.

Response Examples

Success

Status: 201 Created

Failure

Status: 403 Forbidden

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

Status: 404 Not Found

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

Comment

- A boat cannot be assigned to multiple owners.

Boat removed from Owner

Boat has been removed from the Owner. Unprotected endpoint.

```
DELETE /owners/:owner_id/boats/:boat_id
```

Request

Request Parameters

Name	Type	Description	Required?
owner_id	String	ID of the owner	Yes
boat_id	String	ID of the boat	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 an owner exists with this owner_id, a boat exists with this boat_id and this boat is owned by this owner.
Failure	404 Not Found	No boat with this boat_id is owned by the owner with this owner_id.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found
```

```
{
```



```
"Error": "No boat with this boat_id is owned by the owner with this owner_id"
}
```

View Loads for a Boat

Load has been removed from the boat. Unprotected endpoint.

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

Request

Request Parameters

Name	Type	Description	Required?
Boat_id	String	ID of the boat	Yes

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Succeeds only if a boat exists with this boat_id, and a load exists with this load_id.
Failure	404 Not Found	No boat with this boat_id exists.

Response Examples

Success

Status: 200 OK

```
[
{
  "id": "123abc",
```

```
"weight": 20,  
  "content": "Legos",  
  "delivery_date": "10/10/2010",  
  "carrier": "abc123",  
  "self": "https://<your-app>/slips/123abc"  
},  
{  
  "id": "456def",  
  "weight": 200,  
  "content": "Hover bike",  
  "Delivery_date": "10/25/2015",  
  "carrier": null,  
  "self": "https://<your-app>/slips/456def"  
}  
]
```

Failure

Status: 404 Not Found

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

View Boats for an Owner

View the Boats owned by an Owner. Unprotected endpoint.

GET /owners/:owner_id/boats/

Request

Request Parameters

Name	Type	Description	Required?
Boat_id	String	ID of the boat	Yes

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Succeeds only if an owner exists with this owner_id.
Failure	404 Not Found	No owner with this owner_id exists.

Response Examples

Success

Status: 200 OK

```
[
{
"loads": [],
"length": 280,
"name": "testboat",
"self": "http://localhost:8080/boats/4817117325033472",
```

```
"type": "yacht",  
"id": "4817117325033472",  
"owner": {  
  "id": "5163227868561408",  
  "user_id": "105294518571022308998",  
  "self": "http://localhost:8080/owners/5163227868561408"  
}  
]
```

Failure

```
Status: 404 Not Found  
  
{  
  "Error": "No owner with this owner_id exists"  
}
```