

Marina Dockyard API Documentation

Table of Contents

Title Page	1
Data Model	3
Create a Boat	4
View a Boat	6
View all Boats	7
Delete a Boat	8
Create a Load	9
View a Load	11
Delete a Load	13
Remove a Load from a Boat	14
View all Loads for a given Boat	15

Data Model

The app stores two kinds of entities in Datastore, Boats and Loads.

User Entity

Property	Data Type	Notes
id	Integer	The id of the user. Datastore automatically generates it.
email	String	User email.
self	String	Link to user profile.
sub	List	Identifies the principal that is the subject of the JWT.

Non-User Entities:

Boats

Property	Data Type	Notes
id	Integer	The id of the boat. Datastore automatically generates it.
name	String	Name of the boat.
type	String	Type of the boat. E.g., Sailboat, Catamaran, etc.
loads	List	List of loads assigned to Boat. Loads to Boat is a M:1 Relationship.
length	Integer	The length of the boat in feet.
owner	String	Lists user id of entity's owner.
public	String	Determines whether an entity is public or private.
self	String	Self link that points to the most canonical representation of that entity instance.

Loads

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	List	List of attributes for the boat the load is assigned to.
item	String	Description of load.
creation_date	String	Date the load was created.
owner	String	Lists user id of entity's owner.
public	String	Determines whether an entity is public or private.
self	String	Self link that points to the most canonical representation of that entity instance.

Create a Boat

Allows you to create a new boat.

POST /boats

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

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
loads	List of loads assigned to the boat.	No

Request Body Example

```
{
  "id": 123,
  "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 is not created, and 400 status code is returned. It is assumed that if the request contains any of the listed attributes, then that attribute's value is valid. It is assumed the request will not contain any extraneous attribute.

Response Examples

Datastore will automatically generate an ID and store it with the entity being created. App sends back this value in the response body as shown in the example.

Success

Status: 201 Created

```
{
  "id": 1234,           # automatically generated by Datastore
  "name": "Sea Witch",  # The name of the boat, a string
  "type": "Catamaran", # The type of boat, power boat, sailboat, catamaran etc.
  "length": 28,         # The length of the boat, an integer
  "loads": [],          # This embedded list does NOT need to paginate.
  "self": "https://appspot.com/boats/1234"
}
```

Failure

Status: 400 Bad Request

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

Note

self	string	Self link goes to the most direct location to find the boat resource. Self attribute is generated and not stored in Datastore.
------	--------	---

View a Boat

Allows you to get an existing boat

GET /boats/:boat_id

Request

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	404 Not Found	No boat with this boat_id exists
Failure	401 Error	Invalid/Missing JWT
Failure	405 Method not authorized	
Failure	406 Invalid Accept Header	

Response Examples

Success

Status: 200 OK

```
{ "id": 1234,          # automatically generated by Datastore
  "name": "Sea Witch", # The name of the boat, a string
  "type": "Catamaran", # The type of boat, power boat, etc. a string
  "length": 28,        # The length of the boat, an integer
  "loads": [
    {"id": 5678,
     "self": "https://appspot.com/loads/5678"},
    {"id": 8765,
     "self": "https://appspot.com/loads/8765"}
  ], # This embedded list does NOT need to paginate.
  "owner": "105614307733648179845"
  "public": "false"
  "self": "https://appspot.com/boats/1234" }
```

Failure

Status: 404 Not Found

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

View all Boats

List all the boats. Implements pagination. Returns 5 boats per page.

Displays a link to get the next page of results, unless there are no more pages of result (i.e., the last page of results doesn't have the next link).

GET /boats

Request

Path 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": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28
  "Loads": [],
  "owner": "105614307733648179845"
  "public": "false"
  "self": "https://appspot.com/boats/1234"
},
{
  "id": 456,
  "name": "Adventure",
  "type": "Sailboat",
  "length": 50
  "Loads": [],
  "owner": "105614307733648179845"
  "public": "false"
  "self": "https://appspot.com/boats/1235"}
,
total: 2]
```

Delete a Boat

Allows you to delete a boat. Deleting a boat will unload any loads that were loaded onto it.

DELETE /boats/:boat_id

Request

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	404 Not Found	No boat with this boat_id exists
Failure	401 Error	Invalid/Missing JWT
Failure	405 Method not authorized	
Failure	406 Invalid Accept Header	

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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

Status: 401 Error

```
{"Error": "Invalid JWT"}
```

Create a Load

Allows you to create a new load. All newly created loads 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
carrier	The boat this load is assigned to.	No
item	Description of load.	Yes
creation_date	Date the 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 is missing the volume, item, or creation_date attribute, the load is not created, and 400 status code is returned. - Doesn't validate the values of this attribute and assumes that if the attribute is specified, then its value is valid. - Assumes that the request does not contain any extraneous attribute
Failure	405 Method not authorized	

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 `carrier` is a list of the name and id of the boat this load is assigned to. Since all newly created loads begin unassigned to any boat, the value of `carrier` should be null.

Success

Status: 201 Created

```
{ "id": 5678, # automatically generated by the Datastore
  "volume": 5,      # The volume of the load, an integer
  "carrier": {},    # The boat carrying the load
  "item": "LEGO Blocks",
  "creation_date": "10/18/2020" # Date the load was created
  "owner": "105614307733648179845"
  "public": "false"
  "self": "https://appspot.com/loads/5678"
}
```

Failure

Status: 400 Bad Request

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

View a Load

Allows you to get an existing load.

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 this load_id exists
Failure	401 Error	Invalid/Missing JWT
Failure	405 Method not authorized	

Response Examples

Success

Status: 200 OK

```
{ "id": 5678, # automatically generated by the Datastore
  "volume": 5, # The volume of the load, an integer
  "carrier": {
    "id": 1234,
    "name": "Sea Witch",
    "self": "https://appspot.com/boats/1234"
  }, # The boat carrying the load
  "item": "LEGO Blocks",
  "creation_date": "10/18/2020" # Date the load was created
  "owner": "105614307733648179845"
  "public": "false"
  "self": "https://appspot.com/loads/5678" }
```

Failure

Status: 404 Not Found

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

Status: 401 Error

```
{ "Error": "Invalid JWT" }
```

View all Loads

List all the loads. Implements pagination similar to view all boats, i.e., 3 loads per page and a next link.

GET /loads

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	405 Method not authorized	
Failure	401 Error	Invalid/Missing JWT

Response Examples

Success

Status: 200 OK

```
[{ "id": 5678, # automatically generated by the Datastore
  "volume": 5,      # The volume of the load, an integer
  "carrier": {
    "id": 1234,
    "name": "Sea Witch",
    "self": "https://appspot.com/boats/1234"}, # The boat carrying the load
  "item": "LEGO Blocks",
  "creation_date": "10/18/2020" # Date the load was created
  "owner": "105614307733648179845"
  "public": "false"
  "self": "https://appspot.com/loads/5678"
},
{ "id": 5679, # automatically generated by the Datastore
  "volume": 7,      # The volume of the load, an integer
  "carrier": {
    "id": 1235,
    "name": "Lucia",
    "self": "https://appspot.com/boats/1235"}, # The boat carrying the load
  "item": "Furnitures",
  "creation_date": "10/18/2020" # Date the load was created
  "owner": "105614307733648179845"
  "public": "false"
  "self": "https://appspot.com/loads/5678"
}];
Total: 2]
```

Delete a Load

Allows you to delete a load. Deleting a load updates 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
Failure	401 Error	Invalid/Missing JWT
Failure	405 Method not authorized	

Response Examples

Success

Status: 204 No Content

Failure

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

```
Status: 401 Not Found
{"Error":  "Invalid JWT" }
```

Assign a Load to a Boat

A load can only be on one boat, but a boat can have many pieces of load.

PUT /boats/:boat_id/loads/:load_id

Request

Path Parameters

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

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 unassigned.
Failure	403 Forbidden	The load is already assigned to another boat and must be removed first.
Failure	401 Error	Invalid/Missing JWT
Failure	401 Error	The boat/load with this id is owned by someone else.
Failure	404 Not Found	The specified boat and/or load does not exist
Failure	405 Method not authorized	

Response Examples

Success

Status: 204 No Content

Failure

Status: 403 Forbidden

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

Status: 404 Not Found

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

Status: 401 Not Found

```
{"Error": "Invalid JWT" }
```

Remove a Load from a Boat

Removes a load from a boat without deleting that load.

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

Request

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

Outcome	Status Code	
Success	204	No Content
Failure	404	Not Found
Failure	401 Error	Invalid/Missing JWT
Failure	405 Method not authorized	

Response Examples

Success

```
Status: 204 No Content
```

Failure

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

View all Loads for a given Boat

List all the loads for a given boat without pagination.

GET /boats/boat_id/loads

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Error	Invalid/Missing JWT
Failure	405 Method not authorized	

Response Examples

Success

Status: 200 OK

```
[{ "id": 5678, # automatically generated by the Datastore
  "volume": 5,      # The volume of the load, an integer
  "carrier": {
    "id": 1234,
    "name": "Sea Witch",
    "self": "https://appspot.com/boats/1234"}, # The boat carrying the load
  "item": "LEGO Blocks",
  "creation_date": "10/18/2020" # Date the load was created
  "self": "https://appspot.com/loads/5678"},
{ "id": 5679, # automatically generated by the Datastore
  "volume": 7,      # The volume of the load, an integer
  "carrier": {
    "id": 1234,
    "name": "Sea Witch",
    "self": "https://appspot.com/boats/1234"}, # The boat carrying the load
  "item": "Furnitures",
  "creation_date": "10/18/2020" # Date the load was created
  "self": "https://appspot.com/loads/5678"}];]
```

Failure

Status: 401 Not Found

```
{"Error": "Invalid JWT" }
```