

Assignment 4: API Spec

CS 493: Cloud Application Development

Fall 2020

Clinton Hawkes

Last Update: Oct 26, 2020. 6:00 pm Pacific

Change log.....	1
Data-Store Model.....	2
Create a Boat.....	3
View a Boat.....	5
View all Boats.....	6
Delete a Boat.....	7
Create a Load.....	8
View a Load.....	10
View all Loads.....	11
View all Loads on Specified Boat.....	12
Delete a Load.....	14
Assign Load to Boat.....	15
Remove Load from Boat.....	16

Change log

Page	Change	Date
	Initial version.	M, Oct 26

APP URL: <https://hw4-hawkes.wl.r.appspot.com>

Data-Store Data Model

Information regarding the relationship between the Boats and Loads is stored in both the entities.

Boats will store all the load id's assigned to it in an array under the attribute labeled "loads". This array may have many id's listed in it or it may be an empty array if there have been no loads assigned to the boat. The three remaining attributes are provided by the user when the entity is created. Information about each attribute is provided in the example below:

Boats Data-store:

"name": "Sea Witch",	***String, name of the boat
"type": "Catamaran",	***String, type of boat
"length": 28,	***Integer, length of the boat
"loads": [1111111,2222222,3333333]	***These are the id's of loads assigned

Loads will store the id of the boat that the load was assigned to. It is stored under the attribute labeled "carrier". This attribute will be null if the load has not been assigned to a boat. It is only permitted to contain a maximum of one integer, so a load may only be assigned to one boat at a time. The three remaining attributes are provided by the user when the entity is created. Information about each attribute is provided in the example below:

Loads Data-store:

"weight": 12,	***Integer, weight of load
"content": "Electronics",	***String, content of load
"delivery_date": "11/12/2020",	***String, date load to be delivered
"carrier": null	***Integer, single id of boat assigned

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

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	<p>If the request is missing any of the 3 required attributes, the boat must not be created, and 400 status code must be returned.</p> <p>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's 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 the ones that are listed).</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 `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
- Loads attribute will contain an empty array. When loads are assigned to the boat, the array will add the id of the load being added.

Success

Status: 201 Created

```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "self": "http://<your-app>/boats/abc123"
}
```

Failure

Status: 400 Bad Request

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

View 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

Response Examples

Success

Status: 200 OK

```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [{
    "id": "xyz321",
    "self": "http://<your-app>/loads/xyz321"
  }],
  "self": "http://<your-app>/boats/abc123"
}
```

Failure

Status: 404 Not Found

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

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
[ "Boats":
  [
    {
      "id": "abc123",
      "name": "Sea Witch",
      "type": "Catamaran",
      "length": 28,
      "loads": [],
      "self": "http://<your-app>/boats/abc123"
    },
    {
      "id": "def456",
      "name": "Adventure",
      "type": "Sailboat",
      "length": 50,
      "loads": [],
      "self": "http://<your-app>/boats/def456"
    },
    {
      "id": "xyz123",
      "name": "Hocus Pocus",
      "type": "Sailboat",
      "length": 100,
      "loads": [],
      "self": "http://<your-app>/boats/xyz123"
    }
  ]
]
```

Delete a Boat

Allows you to delete a boat. Note that if the boat is currently in a load, deleting the boat makes the load empty.

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

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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

Create a Load

Allows you to create a new load.

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	What the load contains	Yes
delivery_date	String	Date the load is scheduled to be delivered	Yes

Request Body Example

```
{
  "weight": 12,
  "content": "Electronics",
  "delivery_date": "11/12/2020"
}
```

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 attributes, the load 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 this load has been assigned. If this load has not been assigned to a boat, the value of `carrier` should be null
- The value of the attribute `self` is a live link to the REST resource corresponding to this load. In other words, this is the URL to get this newly created load. You must not store this attribute in Datastore

Success

Status: 201 Created

```
{
  "id": "123abc",
  "weight": 12,
  "carrier": null,
  "content": "Electronics",
  "delivery_date": "11/12/2020",
  "self": "http://<your-app>/loads/123abc"
}
```

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

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": 12,
  "carrier": { "id": 987654,
    "self": "http://<your-app>/boats/987654" },
  "content": "Electronics",
  "delivery_date": "11/12/2020",
  "self": "http://<your-app>/loads/123abc"
}
```

Failure

Status: 404 Not Found

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

View all Loads

List all the loads.

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

```
[{"Loads": [{"id": "123abc", "weight": 12, "carrier": {"id": 987654, "self": "http://<your-app>/boats/987654"}, "content": "Electronics", "delivery_date": "11/12/2020", "self": "http://<your-app>/loads/123abc"}, {"id": "456def", "weight": 12, "carrier": {"id": 675849, "self": "http://<your-app>/boats/675849"}, "content": "Sports equipment", "delivery_date": "11/15/2020", "self": "http://<your-app>/loads/456def"}]}
```

View all Loads for Specified Boat

List all the loads.

GET /boats/:boat_id/loads

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

Response Examples

Success

Status: 200 OK

```
[{"Loads": [{"id": "123abc", "weight": 12, "carrier": {"id": 987654, "self": "http://<your-app>/boats/987654"}, "content": "Electronics", "delivery_date": "11/12/2020", "self": "http://<your-app>/loads/123abc"}, {"id": "456def", "weight": 12, "carrier": {"id": 987654, "self": "http://<your-app>/boats/987654"}, "content": "Sports equipment", "delivery_date": "11/15/2020", "self": "http://<your-app>/loads/456def"}]}
```

Failure

Status: 404 Not Found

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

Delete a Load

Allows you to delete a load. If the load being deleted has a boat, the boat is now considered “at sea.”

```
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"
}
```

Assign Load to Boat

Boat has arrived at a load.

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

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 has not been assigned.
Failure	403 Forbidden	There is already a boat at this load.
Failure	404 Not Found	No boat with this boat_id exists, and/or no load with this load_id exists.

Response Examples

Success

Status: 204 No Content

Failure

Status: 403 Forbidden

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

Status: 404 Not Found

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

Remove Load from Boat

Boat has left the load to go to sea. The load is now empty.

```
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 boat is at this load.
Failure	404 Not Found	No boat with this boat_id is at the load with this load_id.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found
```

```
{
  "Error": "No boat with this boat_id has a load with this load_id"
}
```