Assignment 4: API Spec

URL:

https://arifig-osu-hw4.ue.r.appspot.com/

CS 493: Cloud Application Development

Spring 2020

Oregon State University

Last Update: April 24, 2020. 9:00 am Pacific

Change log

Page	Change	Date
	Initial version.	Sn, April 12

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
loads	array	Length of the boat in feet.	No

Request Body Example

```
{
  "name": "Sea Witch",
   "type": "Catamaran",
  "length": 28,
   “loads” : [
      {“id”: 12345}
    ]
}
```

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 must not be created, and 400 status code must be returned. 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. 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).

Failure	404 Not Found	If the request contains the optional ‘loads’ parameter, and at least one of the loads in the ‘loads’ array does not exist, we return a 404 Not Found with a message: {"Error": "At least one load with the provided load_ids does not exist"}
Failure	403 Forbidden	If the request contains the optional ‘loads’ parameter and at least one of the boats in the ‘loads’ array is already assigned to a boat (even if it is already assigned to the current boat), then we return a 403 error with the following message: {"Error": "At least one load with the provided load_ids already has a carrier"}

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,
   “loads”: [
      {“id”: 12345,
        “self”: "https://<your-app>/loads/12345" }
    ],
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

```
Status: 400 Bad Request

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

Failure

```
Status: 404 Not Found
```

```
{
"Error": "At least one load with the provided load_ids already has a carrier"
}
```

Failure

```
Status: 403 Forbidden

{
"Error": "At least one load with the provided load_ids does not exist"
}
```

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

Response Examples

Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Sea Witch",
   "type": "Catamaran",
  "length": 28,
   “loads”: [
      {“id”: 12345,
       “self”: "https://<your-app>/loads/12345" }
    ],
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

```
Status: 404 Not Found

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

Get all loads from Boat

Allows you to get an existing boat

```
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

[
  {
    "id": 5112644562321408,
    "self": "http://127.0.0.1:8080/loads/5112644562321408"
  },
  {
    "id": 5647975057457152,
    "self": "http://127.0.0.1:8080/loads/5647975057457152"
  },
  {
    "id": 5707651547660288,
    "self": "http://127.0.0.1:8080/loads/5707651547660288"
  }
]
```

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:
[
•{
•length:1,
•type:"small",
•name:"a boat",
•loads:
[
•{
•self:"http://127.0.0.1:8080/loads/5112644562321408",
•id:5112644562321408
},
•{
•id:5647975057457152,
•self:"http://127.0.0.1:8080/loads/5647975057457152"
},
•{
•id:5707651547660288,
•self:"http://127.0.0.1:8080/loads/5707651547660288"
}
],
•self:"http://127.0.0.1:8080/boats/5066704988143616",
•id:5066704988143616
},
•{
•name:"Odyssey",
•loads: [ ],
•length:99,
•type:"Yatch",
•self:"http://127.0.0.1:8080/boats/5088723339313152",
•id:5088723339313152
},
•{
•length:3,
•type:"small",
•name:"c boat",
•loads:
[
•{
•self:"http://127.0.0.1:8080/loads/5739956077068288",
•id:5739956077068288
}
],
•self:"http://127.0.0.1:8080/boats/5149887163269120",
•id: 5149887163269120
}
],
•next:"http://127.0.0.1:8080/boats?limit=3&offset=3"
}
```

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

None

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	Description of what the load is carrying	Yes
delivery_date	String	Date the load will be delivered	Yes
carrier	Entity	JSON formatted object of the boat carrying the load	No

Request Body Example

```
{
      "weight": 7,
      "content": "FFF",
      "delivery_date": "1/21/2020"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing the number attribute, the load must not be created, and 400 status code must be returned. 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. 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).

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 a json object which has a key of “id” that represents the id of the boat currently carrying this load. If there is no boat for 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 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

{
   "carrier": null,
   "content": "FFF",
   "delivery_date": "1/21/2020",
   "id": 5701666712059904,
   "self": "http://127.0.0.1:8080/loads/5701666712059904",
   "weight": 7
}
```

Failure

```
Status: 400 Bad Request

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

Get 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

{
  "carrier": {
    "id": 5710150413320192,
    "name": "f boat",
    "self": "http://127.0.0.1:8080/boats/5710150413320192"
  },
  "content": "FFF",
  "delivery_date": "1/21/2020",
  "id": 5701666712059904,
  "self": "http://127.0.0.1:8080/loads/5701666712059904",
  "weight": 7
}
```

Failure

```
Status: 404 Not Found

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

List 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:
[
•{
•delivery_date:"1/21/2020",
•content:"CCC",
•carrier:
{
•id:5066704988143616,
•self:"http://127.0.0.1:8080/boats/5066704988143616"
},
•weight:4,
•self:"http://127.0.0.1:8080/loads/5112644562321408",
•id:5112644562321408
},
•{
•content:"test load",
•weight:70,
•carrier:null,
•delivery_date:"1/21/2020",
•self:"http://127.0.0.1:8080/loads/5118200840716288",
•id:5118200840716288
},
•{
•content:"AAA",
•weight:2,
•carrier:null,
•delivery_date:"1/21/2020",
•self:"http://127.0.0.1:8080/loads/5175759811379200",
•id:5175759811379200
}
],
•next:"http://127.0.0.1:8080/loads?limit=3&offset=3"
}
```

Delete a Load

Allows you to delete a load. If the load being deleted has a boat, the load from that boats “loads” array is also removed.

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

Request

Request Parameters

None

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"
}
```

Add a Load to a Boat

Adds a load to a boat.

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

Request

Request Parameters

None

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 empty.
Failure	403 Forbidden	There is already a boat assigned to this load.
Failure	404 Not Found	No boat with this boat_id exists, and/or no load with this load_id exits.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 403 Forbidden

{
"Error": "The load is already assigned to a ship"

}
```

```
Status: 404 Not Found

{
"Error":  "The specified boat and/or load don’t exist"
}
```

Comment

- A load cannot be assigned to multiple boats.

Remove Load from Boat

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

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

Request

Request Parameters

None

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 loads are assigned to this boat (boat[‘loads’] == null)
Failure	404 Not Found	No boat/load with this boat_id/load_id exists
Failure	404 Not Found	The load_id provided exists, but is not assigned to this boat

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

{
"Error": "No boat assigned to this load"
}
```

```
Status: 404 Not Found

{
"Error": "No load with this load_id exists or boat with this boat_id exists"

}
```

```
Status: 404 Not Found
{
"Error": "load_id not found on this boat"}
```