Assignment 4: API Spec

Jesse Piccione

CS 493: Cloud Application Development

Spring 2023

Oregon State University

Endpoint: https://irapi-385318.uc.r.appspot.com

Change log

Version	Change	Date
1.0	Initial version.	5/1/23

Data Model

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

Boats

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.
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	Array of json objects containing ids for loads
self	String	A link to to the get request of the entity's self

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
item	String	The name of the item in the load
creation_date	String	The date the load was created
carrier	JSON Object	JSON object containing (id, name, self) for carrier(boat) of load
self	String	Link to get request for load

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

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	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 that is not listed above).

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Success

```
Status: 201 Created

{
   "name": "Sea Witch",
   "length": 28,
   "loads": [],
   "self": "http://127.0.0.1:8080/boats/5747559041597440",
   "type": "Catamaran",
   "id": "5747559041597440"
}
```

Failure

```
Status: 400 Bad Request

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

Get a Boat

Allows you to get an existing boat

GET /boats/:id

Request

Path Parameters

Name	Description
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

Response Examples

Success

```
Status: 200 OK
{
   "name": "Sea Witch",
   "length": 28,
   "loads": [],
   "self": "http://127.0.0.1:8080/boats/5747559041597440",
   "type": "Catamaran",
   "id": "5747559041597440"
}
```

Failure

```
Status: 404 Not Found

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

List all Boats

List all the boats.

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


```
{
   "boats": [
      {
         "name": "401K",
         "length": 18,
         "loads": [],
         "self": "http://127.0.0.1:8080/boats/5081834211770368",
         "type": "Yacht"
      },
      {
         "name": "Lucia",
         "length": 20,
         "loads": [],
         "self": "http://127.0.0.1:8080/boats/5082649047597056",
         "type": "Catamaran"
      },
      {
         "name": "Lucia",
         "length": 20,
         "loads": [],
         "self": "http://127.0.0.1:8080/boats/5087502092206080",
         "type": "Catamaran"
      }
   ]
```

```
}
```

Delete a Boat

Allows you to delete a boat. Note that if the boat has a load, deleting the boat removes all loads from the boat.

DELETE /boats/:id

Request

Path Parameters

Name	Description
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

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

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	Volume of the load	Yes
item	Name of the item in the load	Yes
creation_date	The date the load was created	Yes

Request Body Example

```
{
  "volume": 3,
  "item": "Food",
  "creation_date": "10/19/2021"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing the 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 the ID of the boat currently holding this load. If there is no boat, the value of `carrier` should be null.

Success

Status: 201 Created


```
{
   "volume": 5,
   "item": "LEGO Blocks",
   "carrier": null,
   "self": "http://127.0.0.1:8080/loads/5669937037180928",
   "creation_date": "10/18/2021",
   "id": "5669937037180928"
}
```


Failure

Status: 400 Bad Request


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

Get a Load

Allows you to get an existing load.

GET /loads/:id

Request

Path Parameters

Name	Description
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

Response Examples

Success

Status: 200 OK


```
{
   "volume": 5,
   "item": "LEGO Blocks",
   "carrier": null,
   "self": "http://127.0.0.1:8080/loads/5669937037180928",
   "creation_date": "10/18/2021",
   "id": "5669937037180928"
}
```


Failure

Status: 404 Not Found


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

List all Loads

List all the loads.

GET /loads

Request

Path Parameters

Name	Description
page	page of the list of loads

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success


```
{
   "loads": [
      {
         "volume": 5,
         "item": "LEGO Blocks",
         "loads": [],
         "self": "http://127.0.0.1:8080/loads/5087093063680000",
         "creation_date": "10/18/2021"
      },
      {
         "volume": 3,
         "item": "Food",
         "carrier": {
            "name": "Lucifer",
            "self": "http://127.0.0.1:8080/boats/5753354428874752",
            "id": "5753354428874752"
         },
         "self": "http://127.0.0.1:8080/loads/5088344509775872",
         "creation_date": "10/19/2021"
      },
      {
         "volume": 4,
         "item": "Toys",
         "carrier": {
```

```
            "name": "Lucifer",
            "self": "http://127.0.0.1:8080/boats/5638246889422848",
            "id": "5638246889422848"
         },
         "self": "http://127.0.0.1:8080/loads/5096936860286976",
         "creation_date": "10/21/2021"
      }
   ]
}
```

Delete a load

Allows you to delete a load. If the load being deleted has a boat, that load is removed from that boat

DELETE /loads/:id

Request

Path Parameters

Name	Description
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"
}
```

Load Arrives at a Boat

Load has arrived by boat .

PUT /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

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 when boat is given a valid load
Failure	403 Forbidden	This load was assigned to another boat already
Failure	404 Not Found	The specified boat and/or load does not exist

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"
}
```

Comment

- A boat can have infinite loads.
- A load can only have one boat.

Load leaves a Boat

A load can be taken off a boat

```
DELETE /boats/:boats_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	Notes
Success	204 No Content	Succeeds only if a boat contains the specified load
Failure	404 Not Found	No boat was found with gived boat_id Or No load was found with given load_id Or No boat was found with given load_id

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'
}
```