

Assignment 4: API Spec

CS 493: Cloud Application Development

Fall 2020

Oregon State University

GCP Link: <https://assignment-3-hg.wl.r.appspot.com/>

Contents

Change log.....	3
Create a Boat	4
Request	4
Request Parameters.....	4
Request Body	4
Request Body Format.....	4
Request JSON Attributes.....	4
Request Body Example.....	4
Response	4
Response Body Format	4
Response Statuses	4
Response Examples.....	4
Get a Boat	6
Request	6
Request Parameters.....	6
Request Body	6
Response	6
Response Body Format	6
Response Statuses	6
Response Examples.....	6
List all Boats	7
Request	7
Request Parameters.....	7
Request Body	7
Response	7
Response Body Format	7

Response Statuses	7
Response Examples.....	7
Delete a Boat.....	9
Request	9
Request Parameters.....	9
Request Body	9
Response	9
Response Body Format	9
Response Statuses	9
Response Examples.....	9
Create a Load	10
Request	10
Request Parameters.....	10
Request Body	10
Request Body Format.....	10
Request JSON Attributes	10
Request Body Example.....	10
Response	10
Response Body Format	10
Response Statuses	10
Response Examples.....	11
Get a Load	12
Request	12
Request Parameters.....	12
Request Body	12
Response	12
Response Body Format	12
Response Statuses	12
Response Examples.....	12
List all Loads	13
Request	13
Request Parameters.....	13
Request Body	13

Response	13
Response Body Format	13
Response Statuses	13
Response Examples.....	13
Delete a Load	15
Request	15
Request Parameters.....	15
Request Body	15
Response	15
Response Body Format	15
Response Statuses	15
Response Examples.....	15
Request	16
Request Parameters.....	16
Request Body	16
Response	16
Response Body Format	16
Response Statuses	16
Response Examples.....	16
Comment	17
Remove Load from Boat	18
Request	18
Request Parameters.....	18
Request Body	18
Response	18
Response Body Format	18
Response Statuses	18
Response Examples.....	18

Change log

Page	Change	Date
	Initial version.	M, Oct 12
	Forked from initial version to fit Assignment 4 API.	T, Oct 27

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

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"
}
```

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": 4862202200719360,
  "length": 99,
  "loads": [],
  "name": "Odyssey",
  "self": "http://127.0.0.1:8080/boats/4862202200719360",
  "type": "Yatch"
}
```

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	Self must be added to the response for each load and boat. Pagination is required. At most 3 items per page.

Response Examples

Success

Status: 200 OK

```
{
  "boats": [
    {
      "id": 4862202200719360,
      "length": 99,
      "loads": [],
      "name": "Odyssey",
      "self": "http://127.0.0.1:8080/boats/4862202200719360",
      "type": "Yatch"
    },
    {
      "id": 5082810578632704,
      "length": 99,
      "loads": [
        {
          "id": 5750704903815168,
          "self": "http://127.0.0.1:8080/loads/5750704903815168"
        }
      ],
      "name": "Odyssey",
      "self": "http://127.0.0.1:8080/boats/5082810578632704",
    }
  ]
}
```

```
    "type": "Yatch"
  },
  {
    "id": 5087502092206080,
    "length": 99,
    "loads": [
      {
        "id": 5675594515742720,
        "self": "http://127.0.0.1:8080/loads/5675594515742720"
      }
    ],
    "name": "Odyssey",
    "self": "http://127.0.0.1:8080/boats/5087502092206080",
    "type": "Yatch"
  }
],
"next": "http://127.0.0.1:8080/boats?limit=3&offset=3"
}
```


Delete a Boat

Unload and delete a boat.

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	All loads on the boat must be unloaded before deleting the boat.
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 /load

Request

Request Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
weight	Integer	Weight of the load.	Yes
content	String	Description of the load.	Yes
deliver_date	String	The date the load is to be delivered	Yes

Request Body Example

```
{
  "weight": 5,
  "content": "LEGO Blocks",
  "delivery_date": "10/02/2020"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	<p>If the request is missing the number attribute, 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

Success

Status: 201 Created

```
{
  "carrier": null,
  "content": "LEGO Blocks",
  "delivery_date": "10/02/2020",
  "id": 5095696621371392,
  "self": "http://127.0.0.1:8080/loads/5095696621371392",
  "weight": 5
}
```

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/: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": null,
  "content": "LEGO Blocks",
  "delivery_date": "10/02/2020",
  "id": "6192604894986240",
  "self": "http://127.0.0.1:8080/loads/6192604894986240",
  "weight": 5
}
```

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	Self must be added to the response for each load and boat (carrier). Pagination is required. At most 3 items per page.

Response Examples

Success

```
{
  "loads": [
    {
      "carrier": {
        "id": 5760334488928256,
        "name": "boat1",
        "self": "http://127.0.0.1:8080/boats/5760334488928256"
      },
      "content": "Load4",
      "delivery_date": "10/02/2020",
      "id": 5087093063680000,
      "self": "http://127.0.0.1:8080/loads/5087093063680000",
      "weight": 5
    },
    {
      "carrier": null,
      "content": "LEGO Blocks",
      "delivery_date": "10/02/2020",
      "id": 5095696621371392,
      "self": "http://127.0.0.1:8080/loads/5095696621371392",
      "weight": 5
    },
    {

```

```
    "carrier": null,  
    "content": "LEGO Blocks",  
    "delivery_date": "10/02/2020",  
    "id": 5102723456303104,  
    "self": "http://127.0.0.1:8080/loads/5102723456303104",  
    "weight": 5  
  },  
],  
"next": "http://127.0.0.1:8080/loads?limit=3&offset=3"  
}
```

Delete a Load

Allows you to delete a load. If the load is on a boat, it must be unloaded before deletion.

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"
}
```

Assign a Load to a Boat.

PUT /boats/:boat_id/:load_id

Request

Request Parameters

Name	Type	Description	Required?
load_id	String	ID of the load	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 a boat exists with this boat_id, a load exists with this load_id.
Failure	403 Forbidden	Load already assigned to a 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: 204 No Content

Failure

Status: 403 Forbidden

```
{
  "Error": "Load already assigned to boat"
}
```

Status: 404 Not Found

```
{
  "Error": "No boat/load with this id exists"
}
```


Comment

- A load cannot be assigned to multiple boats.
- It should be possible to remove a load from a boat without deleting that load.

Remove Load from Boat

Unload a single item from boat.

DELETE /boats/:boat_id/:load_id

Request

Request Parameters

Name	Type	Description	Required?
load_id	String	ID of the load	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 a boat exists with this boat_id, a load exists with this load_id.
Failure	403 Forbidden	Fails if load does is not on 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: 204 No Content

Failure

Status: 403 Forbidden

```
{
  "Error": "This load is not on the boat"
}
```

Status: 404 Not Found

```
{
  "Error": "No boat/load with this id exists"
}
```

Data Model

Property Name	Notes
id	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

Property Name	Notes
id	The id of the load. Datastore automatically generates it. Don't add it yourself as a property of the entity.
weight	Int. Load weight.
content	String. Description of the load.
delivery_date	String. The date the load is to be delivered