Assignment 4: API Spec

CS 493: Cloud Application Development

Fall 2021

Oregon State University

Last Update: Oct 25, 2021. 11:15 am Pacific

[Change log 1](#_Toc85539399)

[Data Model 2](#_Toc85539400)

[Create a Boat 3](#_Toc85539401)

[Get a Boat 5](#_Toc85539402)

[List all Boats 6](#_Toc85539403)

[Delete a Boat 7](#_Toc85539404)

[Get all Loads for a given Boat 8](#_Toc85539405)

[Create a Load 9](#_Toc85539406)

[Get a Load 11](#_Toc85539407)

[List all Loads 12](#_Toc85539408)

[Delete a Load 14](#_Toc85539409)

[Assign a Load to a Boat 15](#_Toc85539410)

[Remove a Load from a Boat 16](#_Toc85539411)

# Change log

|  |  |  |
| --- | --- | --- |
| **Version** | **Change** | **Date** |
| 1.0 | Initial version. | Oct 25, 2021 |

# 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 | The load\_id numbers of the loads the boat carries, if any |

## 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 | Object | The boat carrying the load. An object with the boat id, boat name, and boat URL. Otherwise, null. |
| content | String | Description of the load contents |
| creation\_date | String? | Date the load was created |

# 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. Your app must send back this value 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": 123,  "name": "Sea Witch",  "type": "Catamaran",  "length": 28,  "loads": [],  "self": "https://<your-app>/boats/123"  } |

#### 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

## 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 |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  {  "id": 123,  "name": "Sea Witch",  "type": "Catamaran",  "length": 28,  "loads": [{"id": 5678, "self": "https://appspot.com/loads/5678}],  "self": "https://<your-app>/boats/123"  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  {  "Error": "No boat with this boat\_id exists"  } |

# List all Boats

List all the boats. Implements pagination i.e. 3 boats per page with a ‘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  {"boats": [  {  "id": 123,  "name": "Sea Witch",  "type": "Catamaran",  "length": 28,  "loads": [{"id": 5678, "self": "https://appspot.com/loads/5678}],  "self": "https://<your-app>/boats/123"  },  {  "id": 456,  "name": "Adventure",  "type": "Sailboat",  "length": 50,  "loads": [],  "self": "https://<your-app>/boats/456"  },  {  "id": 789,  "name": "Hocus Pocus",  "type": "Sailboat",  "length": 100,  "loads": [{"id": 1011, "self": "https://appspot.com/loads/1011}],  "self": "https://<your-app>/boats/789"  }],  "next": "https://<your-app>/boats?cursor=cJQslMoWbXsjcsQsM"  } |

# Delete a Boat

Allows you to delete a boat. Note that if the boat currently holds one or many loads, they will be removed from the boat.

|  |
| --- |
| 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 |

### Response Examples

#### Success

|  |
| --- |
| Status: 204 No Content |

#### Failure

|  |
| --- |
| Status: 404 Not Found  {  "Error": "No boat with this boat\_id exists"  } |

# Get all Loads for a given Boat

Allows you to get an existing boat

|  |
| --- |
| GET /boats/:boat\_id/loads |

## 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 |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  {  "loads": [  {"id": 1234, "self": "https://appspot.com/loads/1234},  {"id": 5678, "self": "https://appspot.com/loads/5678},  {"id": 0910, "self": "https://appspot.com/loads/0910}  ]  "self": "https://<your-app>/boats/123/loads"  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  {  "Error": "No boat with this boat\_id exists"  } |

# Create a Load

Allows you to create a new load. All newly created loads are 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 |
| content | Description of the load contents | Yes |
| creation\_date | Date the load was created | Yes |

### Request Body Example

|  |
| --- |
| {  "volume": 5,  "content": "LEGO blocks",  "creation\_date": "10/20/2021"  } |

## 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 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 attributes are specified, then the values are 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. 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 containing this load. If there is no boat carrying 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  {  "id": 123,  "volume": 5,  "carrier": null,  "content": "LEGO blocks",  "creation\_date": "10/20/2021",  "self": "https://<your-app>/loads/123"  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {  "Error": "The request object is missing at least one of the required attributes"  } |

# Get a Load

Allows you to view 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 |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  {  "id": 123,  "volume": 5  "carrier": {  "id": 456,  "name": "Sea Witch",  "self": "https://<your-app>/loads/456"  },  "content": "LEGO blocks",  "creation\_date": "10/20/2021",  "self": "https://<your-app>/loads/123"  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  {  "Error": "No load with this load\_id exists"  } |

# List all Loads

List all the loads. Implements pagination i.e. 3 loads per page with 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 |  |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  {"loads":  [{  "id": 123,  "volume": 5  "carrier": {  "id": 456,  "name": "Sea Witch",  "self": "https://<your-app>/loads/456"  },  "content": "LEGO blocks",  "creation\_date": "10/20/2021",  "self": "https://<your-app>/loads/123"  },  {  "id": 456,  "volume": 10,  "carrier": null,  "content": "Stuff from China",  "creation\_date": "9/20/2021",  "self": "https://<your-app>/loads/456"  },  {  "id": 789,  "volume": 20,  "carrier": null,  "content": "More stuff from China",  "creation\_date": "9/20/2021",  "self": "https://<your-app>/loads/789"  }],  "next": "https://<your-app>/loads?cursor=cJQslMoWbXsjcsQsM"  } |

# Delete a Load

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

|  |
| --- |
| 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 |

### 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

Load has arrived at a Boat. A load cannot be assigned to multiple boats.

|  |
| --- |
| PATCH /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 only if a boat exists with this boat\_id, a load exists with this load\_id, and the load is not already assigned to another boat. |
| Failure | 403 Forbidden | This load is already assigned to a boat. |
| 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 not empty"  } |

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

# Remove a Load from a Boat

Load has been removed from the boat. The boat no longer stores 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** | **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 carrying 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 load with this load\_id is at the boat with this boat\_id"  } |