Assignment 8: API Spec

CS 493: Cloud Application Development

Ashley Owens

Last Update: Nov 17, 2021. 3:30 PM PST

API Link: [http://35.185.222.180:8000](http://35.185.222.180:8000/)

[Change log 1](#_Toc88044820)

[Data Model 2](#_Toc88044821)

[Create a Boat 3](#_Toc88044822)

[Get a Boat 5](#_Toc88044823)

[List all Boats 7](#_Toc88044824)

# Change log

|  |  |  |
| --- | --- | --- |
| **Version** | **Change** | **Date** |
| 1.0 | Initial version. | Nov 17, 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’s that the boat carries, else empty array |

## 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"  } |