CS493 Cloud Development Homework 5 – Spring 2020

Author: Chen Zou

Email: [zouch@oregonstate.edu](mailto:zouch@oregonstate.edu)

------------------------------------------------------------------------------------------------------------------------------

**Application link:** <https://hw5-zouch000.appspot.com/>

Contents

[Change log 1](#_Toc39145387)

[Create a Boat 2](#_Toc39145388)

[Get a Boat 4](#_Toc39145389)

[Edit a Boat - PATCH 6](#_Toc39145390)

[Edit a Boat - PUT 9](#_Toc39145391)

[Delete a Boat 12](#_Toc39145392)

# Change log

|  |  |  |
| --- | --- | --- |
| **Page** | **Change** | **Date** |
|  | Initial version. | April 28, 2020 |
|  |  |  |
|  |  |  |

# 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

|  |
| --- |
| MIME type:  {  "name": "Sea Witch I",  "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, or has extra attributes, the boat must not be created, and 400 status code must be returned.  If the request contains invalid values of the attributes, such as incorrect data type, out of range data, or invalid characters, return 400 as well. |
| Failure | 403 Forbidden | When the name already exists, return forbidden |
| Failure | 406 Not Acceptable | Return 406 If the client sends accept types other than JSON |

### 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 I",  "type": "Catamaran",  "length": 28,  "self": "https://hw4-zouch000.appspot.com/boats/abc123"  } |

#### Failure

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

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {  "Error": "The request object contains unrecognized attributes"  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {  "Error": "The request has invalid content"  } |

#### Failure

|  |
| --- |
| Status: 403 Forbidden  {  "Error": "The name of boat is already taken"  } |

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  {  "Error": "Only JSON is supported as requested content type"  } |

# 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 / HTML according to Accept header from client

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK |  |
| Failure | 404 Not Found | No boat with this boat\_id exists |
| Failure | 406 Not Acceptable | Return 406 If the client sends accept types other than JSON and HTML |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  MIME type: text/html  {  <li>id": "abc123",  <li>"name": "Sea Witch I",  "type": "Catamaran",  "length": 28,  "self": "https:// hw4-zouch000.appspot.com/boats/abc123"  } |

#### Success

|  |
| --- |
| Status: 200 OK  MIME type: Application/JSON  {  "id": "abc123",  "name": "Sea Witch",  "type": "Catamaran",  "length": 28,  "self": "https:// hw4-zouch000.appspot.com/boats/abc123"  } |

#### Failure

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

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  {  "Error": "Only JSON and HTML are supported as return content type"  } |

# Edit a Boat - PATCH

Allows you to edit some attributes of a boat.

|  |
| --- |
| PATCH /boats/:boat\_id |

## Request

### Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| boat\_id | String | ID of the boat | Yes |

### Request Body

Required

### Request Body Format

JSON

### Request JSON Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| name | String | The name of the boat. | \* |
| type | String | The type of the boat. E.g., Sailboat, Catamaran, etc. | \* |
| length | Integer | Length of the boat in feet. | \* |

\* only at least one of the attributes required by PATCH method.

### Request Body Example

|  |
| --- |
| {  "name": "new name",  "length": 999  } |

## Response

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK |  |
| Failure | 400 Bad Request | If the request has no attribute or attributes other than the required 3, the boat must not be updated, and 400 status code must be returned.  If the request contains invalid values of the attributes, such as incorrect data type, out of range data, or invalid characters, return 400 as well. |
| Failure | 403 Forbidden | When client try to edit the ID, or when client trying to update with an unavailable name, return forbidden. |
| Failure | 404 Not Found | No boat with this boat\_id exists. |
| Failure | 406 Not Acceptable | Return 406 If the client sends accept types other than JSON |

### Response Examples

* The self attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

#### Success

|  |
| --- |
| Status: 200 OK  {  “id”: “abc123”,  “name”: “new name”,  “type”: “Catamaran”,  “length”: 999,  “self”: “https://<your-app>/boats/abc123”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {  “Error”: “The request object needs at least one of the attributes”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {  "Error": "The request object contains unrecognized attributes"  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {  "Error": "The request has invalid content"  } |

#### Failure

|  |
| --- |
| Status: 403 Forbidden  {  "Error": "The name of boat is already taken"  } |

#### Failure

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

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  {  "Error": "Only JSON is supported as requested content type"  } |

# Edit a Boat - PUT

Allows you to edit a boat.

|  |
| --- |
| PUT /boats/:boat\_id |

## Request

### Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| boat\_id | String | ID of the boat | Yes |

### 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": "New Name",  "type": "Catamaran new",  "length": 999  } |

## Response

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK |  |
| Success | 303 See Other | Return 303 with the location of updated boat, so the client can make a GET request to the location specified in the Location header of the response. |
| Failure | 400 Bad Request | If the request is missing any of the 3 required attributes, or has extra attributes, the boat must not be updated, and 400 status code must be returned.  If the request contains invalid values of the attributes, such as incorrect data type, out of range data, or invalid characters, return 400 as well. |
| Failure | 403 Forbidden | When client try to edit the ID, or when client trying to update the boat with an unavailable name, return forbidden. |
| Failure | 404 Not Found | No boat with this boat\_id exists. |
| Failure | 406 Not Acceptable | Return 406 If the client sends accept types other than JSON |

### Response Examples

* The self attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

#### Success

|  |
| --- |
| Status: 200 OK  {  “id”: “abc123”,  “name”: “Sea Witch”,  “type”: “Catamaran”,  “length”: 99,  “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: 400 Bad Request  {  "Error": "The request object contains unrecognized attributes"  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {  "Error": "The request has invalid content"  } |

#### Failure

|  |
| --- |
| Status: 403 Forbidden  {  "Error": "The name of boat is already taken"  } |

#### Failure

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

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  {  "Error": "Only HTML is supported as requested content type"  } |

# Delete a Boat

Allows you to delete a boat.

|  |
| --- |
| DELETE /boats/:boat\_id |

## Request

### Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Required?** |
| 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 |  |
| 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"  } |

# Edit All Boats

Unallowed when send PUT requests on the root boat URL.

|  |
| --- |
| PUT /boats |

## Response

No body

### Response Body Format

Failure: JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Failure | 405 Method Not Allowed | When sending PUT to the root boat URL without specific boat ID, return 405 since deleting all boats is not allowed. |

### Response Examples

#### Failure

|  |
| --- |
| Status: 405  {  "Error": "Editing all boats is not allowed"  } |

# Delete All Boats

Unallowed when send DELETE requests on the root boat URL.

|  |
| --- |
| DELETE /boats |

## Response

No body

### Response Body Format

Failure: JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Failure | 405 Method Not Allowed | When sending DELETE to the root boat URL without specific boat ID, return 405 since deleting all boats is not allowed. |

### Response Examples

#### Failure

|  |
| --- |
| Status: 405  {  "Error": "Deleting all boats is not allowed"  } |