https://a5-loyami.uw.r.appspot.com/

Assignment 5: Advanced REST Features

CS 493: Cloud Application Development

Michelle Alexandra Loya

Data Model

The app stores the entity Boats in Datastore.

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.
self	String	The direct URL or link to access the boat. Automatically generated upon creation.

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 name attribute is not valid length (less than 20 characters) or type attribute contains special characters, input is invalid, and 400 status should be returned
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.
Failure	400 Bad Request	If the request contains attributes that are not supported, a 400 status code must be returned.
Failure	403	If the requested boat name is not unique, the boat must not be created, and 403 status code must be returned.
Failure	406	If the request header includes an Accept MIME type that is not supported, the boat must not be created, and 406 status code must be returned.
Failure	415	If the client sends an unsupported MIME type to the endpoint, the boat must not be created, and 415 status code must be returned.

Response Examples

Datastore will automatically generate an ID and store it with the entity being created. A self-attribute URL must be generated on the fly and sent back with all the values in the response body as shown in the example.

Success

```
Status: 201 Created
{
   "id": 123,
   "name": "Sea Witch",
    "type": "Catamaran",
   "length": 28,
    "self": "http://example.com/boats/123"
}
```

Failure

```
Status: 400 Bad Request

{

"Error": {

"name": "Name cannot be longer than 20 characters.",

"type": "Type cannot contain special characters."

}

}
```

```
Status: 400 Bad Request
{
"Error":  " The request object contains unsupported attributes
"
}
```

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

```
Status: 403
{
"Error":  " Boat name must be unique "
}
```

```
Status: 406
{
"Error":  "Accept MIME type not supported "
}
```

```
Status: 415
{
"Error":  "Request must be JSON"
}
```

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, HTML

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No boat with this boat_id exists
Failure	406	If the request header includes an Accept MIME type that is not supported, the requested boat must not be returned, and 406 status code must be returned.

Response Examples

Self-attributes are not stored in Datastore and must generated on the fly (for both boats and loads).

Success

```
Status: 200 OK
{
   "id": 123,
   "name": "Sea Witch",
    "type": "Catamaran",
   "length": 28,
    "self": "http://example.com/boats/123"
}
```

Failure

```
Status: 404 Not Found
{
"Error":  "No boat with this boat_id exists"
}
```

```
Status: 406
{
"Error":  "Accept MIME type not supported "
}
```

List all Boats

List all the boats.

```
GET /boats
```

Request

Path Parameters

None

Query Parameters

Name	Data Type	Description
offset	integer	Specifies where in the list of boats to start returning results. Default is 0.

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406	If the request header includes an Accept MIME type that is not supported, the list of boats must not be returned, and 406 status code must be returned.

Response Examples

The API supports pagination, where a user can specify an offset query parameter to determine the starting point for the list of boats returned. The default return limit is set to 3. If there are more boats available than the limit, a "next" link will be provided in the response to fetch the next set of boats. Self-attributes are not stored in Datastore and must generated on the fly (for both boats and loads).

Success

```
Status: 200 OK
{
[
{
   "id": 123, "name": "Sea Witch",
   "type": "Catamaran",
   "length": 28,
   "loads": [],
   "self": "http://localhost:8080/boats/123"
},
{
   "id": 456,
   "name": "Adventure",
   "type": "Sailboat",
   "length": 50, "loads": [],
   "self": "http://localhost:8080/boats/456"
},
{
   "id": 789,
   "name": "Hocus Pocus",
   "type": "Sailboat",
   "length": 100,
   "loads": [],
    "self": "http://localhost:8080/boats/789"
}
],
"next": "http://localhost:8080/boats?offset=3"
}
```

Failure

```
Status: 406
{
"Error":  "Accept MIME type not supported "
}
```

Delete a Boat

Allows you to delete a 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
Failure	406	If the request header includes an Accept MIME type that is not supported, the boat must not be deleted, and 406 status code must be returned.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found
{
"Error":  "No boat with this boat_id exists"
}
```

```
Status: 406
{
"Error":  "Accept MIME type not supported "
}
```

Patch a Boat

Allows updating any subset of attributes while the other attributes remain unchanged.

```
PATCH /boats/:boat_id
```

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	No
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	No
length	Length of the boat in feet.	No

Request Body Example

```
{
   "name": "Sea Witch",
   "length": 28
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200	The requested attributes are successfully updated
Failure	400 Bad Request	If the request contains bad attributes, the boat must not be edited, and 400 status code must be returned.
Failure	403 Forbidden	If the request is includes changing an ID, no edits must be made, and 403 status code must be returned.
Failure	403 Forbidden	If the requested boat name is not unique, the boat must not be edited, and 403 status code must be returned.
Failure	404 Not Found	If the requested boat does not exist, a 404 status code must be returned.
Failure	406	If the request header includes an Accept MIME type that is not supported, the boat must not be edited, and 406 status code must be returned.
Failure	415	If the client sends an unsupported MIME type to the endpoint, the boat must not be edited, and 415 status code must be returned.

Success

Status: 200

```
{
   "id": 123,
   "name": "Sea Witch",
    "type": "Catamaran",
   "length": 28,
    "self": "http://example.com/boats/123"
}
```

Failure

Status: 400 Bad Request
{
"Error": {
"name": "Name cannot be longer than 20 characters.",
"type": "Type cannot contain special characters."
}
}

Status: 400 Bad Request
{
"Error": " The request object contains unsupported attributes
"
}

Status: 403
{
"Error": " Updating the value of id is not allowed
"
}

Status: 403
{
"Error": " Boat name must be unique "
}

Status: 404
{
"Error": " No boat with this boat_id exists"
}

Status: 406
{
"Error": "Accept MIME type not supported "
}

Status: 415
{
"Error": "Request must be JSON"
}

Put a Boat

Updates all of a boat’s attributes.

PUT /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

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”: “sailboat”,
  "length": 28
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200	The requested attributes are successfully updated
Failure	400 Bad Request	If name attribute is not valid length (less than 20 characters) or type attribute contains special characters, input is invalid, and 400 status should be returned
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat must not be edited, and 400 status code must be returned.
Failure	400 Bad Request	If the request contains bad attributes, the boat must not be edited, and 400 status code must be returned.
Failure	403 Forbidden	If the request is includes changing an ID, no edits must be made, and 403 status code must be returned.

Failure	403 Forbidden	If the requested boat name is not unique, the boat must not be edited, and 403 status code must be returned.
Failure	404 Not Found	If the requested boat does not exist, a 404 status code must be returned.
Failure	406	If the request header includes an Accept MIME type that is not supported, the boat must not be edited, and 406 status code must be returned.
Failure	415	If the client sends an unsupported MIME type to the endpoint, the boat must not be edited, and 415 status code must be returned.

Success

Status: 200

```
{
   "id": 123,
   "name": "Sea Witch",
    "type": "Catamaran",
   "length": 28,
    "self": "http://example.com/boats/123"
}
```

Failure

Status: 400 Bad Request
{
"Error": {
"name": "Name cannot be longer than 20 characters.",
"type": "Type cannot contain special characters."
}
}

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

Status: 400 Bad Request
{
"Error": " The request object contains unsupported attributes
"
}

Status: 403 { "Error": " Updating the value of id is not allowed "

}

Status: 403
{
"Error": " Boat name must be unique "
}

Status: 404
{
"Error": " No boat with this boat_id exists"
}

Status: 406
{
"Error": "Accept MIME type not supported "
}

Status: 415
{
"Error": "Request must be JSON"
}