Portfolio Assignment Final Project: API Spec

CS 493: Cloud Application Development

Fall 2022

Oregon State University

Last Update: Dec 3.

https://angelic-surfer-406717.uw.r.appspot.com

Change log

Version	Change	Date
1.0	Initial version.	Oct 6, 2022

Data Model

The app stores two kinds of entities in Datastore, Boats and Slips.

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	url of the specific boat

Slips

Property	Data Type	Notes
id	Integer	The id of the slip. Datastore automatically generates it. Don't add it yourself as a property of the entity.
size	Integer	The number of the slip.
current_boat	Integer	If there is a boat at the slip, then id of that boat. Otherwise null.
price	String	The price of renting the slip (string because of dollar sign)
condition	String	The condition of the slip
self	String	url of the specific slip
slip	Int	Id of the slip the boat is currently in, starts as Null

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	406 Not Acceptable	Accept header is not json
Failure	401 Unauthorized	No auth header or JWT Token invalid

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.

Success

```
Status: 201 Created

{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  “owner”: user id,
  “slip”: Null,
  “self”: url here,
  “id”: 123
}
```

Failure

```
Status: 406 Not Acceptable

{
"Error":  "Accept header is not json"
}
```

```
Status: 401 Bad Request
{
“Error”: “Authorization header is missing”
}
```

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
Failure	406 Not Acceptable	Accept header is not json
Failure	403 Forbidden	User tried to get boat that does not belong to them

Response Examples

Success

```
Status: 200 OK

{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  “owner”: user id,
  “slip”: Null,
  “self”: url here,
  “id”: 123
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No boat with this boat_id exists"
}
```

```
Status: 406 Not Acceptable

{
"Error":  "Accept header is not json"
}
```

```
Status 403 Forbidden
{
“Error”: “action not permitted”
}
```

List all Boats

List all the boats.

```
GET /boats
```

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Displays 5 boats owned by user per page and total boats owned by user at the bottom. The link to the next page of boats is at the top
Failure	406 Not Acceptable	Accept header is not json

Response Examples

Success

```
Status: 200 OK

[
“link to next page here”
{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  “owner”: user id,
  “slip”: Null,
  “self”: url here,
  “id”: 123
}{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  “owner”: user id,
  “slip”: Null,
  “self”: url here,
  “id”: 123
}{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  “owner”: user id,
  “slip”: Null,
  “self”: url here,
  “id”: 123
}{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  “owner”: user id,
  “slip”: Null,
  “self”: url here,
  “id”: 123
}{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  “owner”: user id,
  “slip”: Null,
  “self”: url here,
  “id”: 123
}Total: 8
]
```

```
Status: 406 Not Acceptable
{
“Error”: “Accept header is not json”
```

Edit all attributes of a Boat

Allows you to edit all attributes of a boat.

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.	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 Finch",
  "type": "Catamaran",
  "length": 28,
  “owner”: user id,
  “slip”: Null,
  “self”: url here,
  “id”: 123
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 OK	Full Edit Succesful
Failure	403 Forbidden	If the wrong owner tries to edit the boat it will fail
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 200 OK
```

Failure

```
Status: 403 Forbidden
{
“Error”: “action not permitted”
}
```

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

Edit one attribute of a Boat

Allows you to edit all attributes of a boat.

```
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.	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 Finch",
 }
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 OK	Full Edit Succesful
Failure	403 Forbidden	If the wrong owner tries to edit the boat it will fail
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 200 OK
```

Failure

```
Status: 403 Forbidden
{
“Error”: “action not permitted”
}
```

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

Delete a Boat

Allows you to delete a boat. Note that if the boat is currently in a slip, deleting the boat makes the slip empty.

```
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 Not Acceptable	Accept header is not json
Failure	403 Forbidden	Action not permitted, wrong user

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

{
"Error":  "No boat with this boat_id exists"
}
```

Failure

```
Status: 406

{
"Error":  "Accept header is not json"
}
```

Failure

```
Status: 403

{
"Error":  "action not permitted"
}
```

Delete All Boats Belonging to A User

Allows you to delete a boat. Note that if the boat is currently in a slip, deleting the boat makes the slip empty.

```
DELETE /boats
```

Request

Path Parameters

Name	Description	Notes
limit	Number of boats per page	Not required
offset	Which boat number the list starts with	Not required

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	406 Not Acceptable	Accept header is not json
Failure	403 Forbidden	Action not permitted, wrong user

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 406

{
"Error":  "Accept header is not json"
}
```

Failure

```
Status: 403

{
"Error":  "action not permitted"
}
```

Create a Slip

Allows you to create a new slip.

```
POST /slips
```

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
size	The size of the slip.	No
price	The price of the slip	No
condition	The condition of the slip	No
current_boat	Current boat in slip, null if empty	Yes

Request Body Example

```
{
  “size”: “large”,
   "price": “$200”,
   “condition”: “repairs needed”,
   “current boat”: Null,
   “self”: url here
   “id”: 123
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	406 Bad Header	Wrong accept header

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The value of the attribute `current_boat` is the ID of the boat currently at this slip. If there is no boat at this slip, the value of `current_boat` should be null.

Success

```
Status: 201 Created

{
   “size”: “large”,
   "price": “$200”,
   “condition”: “repairs needed”,
   “current boat”: Null,
   “self”: url here
```

```
  “id”: 123

}
```

Failure

```
Status: 406 Bad Header

{
"Error":  "Accept header is not json"
}
```

Get a Slip

Allows you to get an existing slip.

```
GET /slips/:slip_id
```

Request

Path Parameters

Name	Description
slip_id	ID of the slip

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No slip with this slip_id exists
Failure	406 Not Acceptable	Accept header is not json

Response Examples

Success

```
Status: 200 OK

{
  “size”: “large”,
  "price": “$200”,
  “condition”: “repairs needed”,
  “current boat”: Null,
  “self”: url here
  “id”: 123
}
```

Failure

```
Status: 404 Not Found
```

```
{
"Error":  "No slip with this slip_id exists"
}
```

```
Status: 406 Not Acceptable
{
“Error”: “accept header is not json”
```

List all Slips

List all the slips.

```
GET /slips
```

Request

Path Parameters

Name	Description	Notes
limit	Number of slips per page	Not required
offset	Which slip number the list starts with	Not required

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	Accept header is not json

Response Examples

Success

```
Status: 200 OK
Next url here
[{
“size”: “large”,
  "price": “$200”,
  “condition”: “repairs needed”,
  “current boat”: Null,
  “self”: url here
  “id”: 123
}
{
“size”: “large”,
  "price": “$200”,
  “condition”: “repairs needed”,
  “current boat”: Null,
  “self”: url here
  “id”: 123
}
{
“size”: “large”,
  "price": “$200”,
  “condition”: “repairs needed”,
  “current boat”: Null,
  “self”: url here
  “id”: 123
}
{
“size”: “large”,
  "price": “$200”,
  “condition”: “repairs needed”,
  “current boat”: Null,
  “self”: url here
  “id”: 123
}
{
“size”: “large”,
  "price": “$200”,
  “condition”: “repairs needed”,
  “current boat”: Null,
  “self”: url here
  “id”: 123
}
Total: 7
]
```

Delete a Slip

Allows you to delete a slip. If the slip being deleted has a boat, the boat is now considered “at sea.”

```
DELETE /slips/:slip_id
```

Request

Path Parameters

Name	**Description**
slip_id	ID of the slip

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 slip with this slip_id exists
Failure	403 Forbidden	Wrong user
Failure	406 Not Acceptable	Bad Header

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
"Error":  "No slip with this slip_id exists"
}
```

Status: 403 Forbidden

```
{
"Error":  "Action not permitted"
}
```

Status: 406 Not Acceptable

```
{
"Error":  "Accept header is not json"
}
```

Boat Arrives at a Slip

Boat has arrived at a slip.

PUT /slips/:slip_id/:boat_id

Request

Path Parameters

Name	**Description**
slip_id	ID of the slip
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 slip exists with this slip_id and the slip is empty.
Failure	403 Forbidden	The slip is not empty because there is already a boat at this slip.
Failure	404 Not Found	The specified boat and/or slip does not exist
Failure	403 Forbidden	The boat is owned by someone else
Failure	406 Not Acceptable	Bad Header

Response Examples

Success

Status: 204 No Content

Failure

Status: 403 Forbidden

```
{
"Error":  "The slip is not empty"
}
```

Status: 403 Forbidden

```
{
“Error”: “action not permitted”
}
```

Status: 404 Not Found

```
{
"Error":  "The specified boat and/or slip does not exist"
}
```

Status: 406 Not Acceptable

```
{
“Error”: “ Accept header is not json”
```

Boat Departs a Slip

Boat has left the slip to go to sea. The slip is now empty.

DELETE /slips/:slip_id/:boat_id

Request

Path Parameters

Name	Description

slip_id	ID of the slip
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 slip exists with this slip_id, this boat is at this slip, and the request is made by the owner of the boat.
Failure	404 Not Found	No boat with this boat_id is at the slip with this slip_id. This could be because no boat with this boat_id exists, or because no slip with slip_id exists, or even if both boat_id and slip_id are valid, the boat with this boat_id is not at this slip with this slip_id
Failure	406 Bad Header	Accept header for the delete request is not json
Failure	403 Wrong User	Incorrect user trying to remove boat from slip

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
"Error":  " No boat with this boat_id is at the slip with this slip_id"
}
```

Status: 406 Bad Header

```
{
“Error”: “Accept header is not json”
}
```

Status: 403 Wrong User

```
{
“Error”: “action not permitted”
}
```

Edit all attributes of a Slip

Allows you to edit all attributes of a boat.

PATCH /slips/:slip_id

Request

Path Parameters

Name	Description
slip_id	ID of the slip

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
size	Size of the slip	No
price	Price of the slip	No
condition	Condition of the slip	No
Current boat	Current boat in the slip	No

Request Body Example

```
{
  “size”: “small”,
  "price": “$200”,
  “condition”: “repairs needed”,
  “current boat”: Null,
  “self”: url here
  “id”: 123
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 OK	Full Edit Successful
Failure	403 Forbidden	If the wrong owner tries to edit the boat it will fail
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 200 OK
```

Failure

```
Status: 403 Forbidden
{
“Error”: “action not permitted”
}
```

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

Edit one attribute of a Slip

Allows you to edit all attributes of a boat.

```
PATCH /slips/:slip_id
```

Request

Path Parameters

Name	Description
boat_id	ID of the slip

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
size	Size of the slip	Yes
price	Price of the slip	Yes
condition	Condition of the slip	Yes

Request Body Example

```
{
  "size": "small",
 }
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 OK	Partial Edit Successful
Failure	404 Not Found	No slip with this slip_id exists

Response Examples

Success

```
Status: 200 OK
```

Failure

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

Delete all slips

This method is not allowed but included so the document is comprehensive.

```
DELETE /slips
```

Request

Path Parameters

None

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Failure	405 Method Not Allowed	If a user attempts to delete all slips the they receive a 405 error code because nobody is allowed to delete all slips.
Failure	406 Bad Header	Accept header for the delete request is not json

Response Examples

Failure

```
Status: 406 Bad Header
{
“Error”: “Accept header is not json”
}
```

```
Status: 403 Wrong User
{
“Error”: “action not permitted”
}
```

List all Users

List all the boats.

```
GET /users
```

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Displays all users in the database
Failure	406 Not Acceptable	Accept header is not json

Response Examples

Success

```
Status: 200 OK

{
```