

Matthew Toro

onid: torom

CS 496 Mobile and Cloud Software Development

REST Planning and Implementation

Due Date: 2/4/2018

API Main Page:

silicon-perigee-191721.appspot.com

POST /boat

- Creates a new boat at sea
- Set Header content-type to application/json
- Send request body as JSON and modeled:

```
{  
  "name": "test_boat",  
  "type": "sail boat",  
  "length": 1  
}
```

where 'name' and 'type' are strings and 'length' is an integer. An id for the boat will automatically be generated.

- Requests not modeled like above will be rejected and send a 400 status code
- If successful, returns status code 201

GET /boat

- Returns every boat in the database

GET /boat/{boat_id}

- Returns the boat with the given id
- If the id is not valid, then it will send a 400 status code back

PATCH /boat/{boat_id}

- Modifies the boat with the given id
- If the id is not valid, then it will send a 400 status code back
- Set Header content-type to application/json
- The request body must be JSON and modeled:

```
{  
  "name": "edit",  
  "type": "edit",  
  "length": 1  
}
```

- If the request does not model the above, it will be rejected and send a 400 status code back

- If you do not want to edit a specific item, then send its current value

PUT /boat/{boat_id}

- Puts the specified boat in the requested slip
- If the id is invalid, then it will send a 400 status code
- Set Header content-type to application/json
- Send in the body JSON data modeled:

```
{
  "number": 1,
  "arrival_date": "edit"
}
```

where 'number' is an integer that refers to the specific slip's number and 'arrival_date' is a string signifying the boat's date of arrival in the slip

- If the body does not match the above, then it will be rejected and send a 400 status code
- If the slip requested is currently occupied, then it will send a 403 status code
- Upon success, the slip will be updated with the current boat's url and the arrival date

DELETE /boat/{boat_id}

- Deletes the boat with the given id
- If the id is not valid, then it will send a 400 status code back
- If there was a slip assigned to the boat, the slip is set to empty

POST /slip

- a new empty slip is created
- Creates a new empty slip
- Set Header content-type to application/json
- Send request body as JSON and modeled:

```
{
  "number": 1
}
```

where 'number' is a unique integer. An id for the slip will automatically be generated.

- If the 'number' is not unique, then it will ignore the request and send a 400 status code
- Requests not modeled like above will be rejected and send a 400 status code

GET /slip

- Returns every slip in the database

GET /slip/{slip_id}

- Returns the slip with the given id
- If the id is not valid, then it will send a 400 status code back

DELETE /slip/{slip_id}

- Deletes the slip with the given id
- If the id is not valid, then it will send a 400 status code back
- If there was a boat assigned to the slip, the boat is set to at sea

PATCH /slip/{slip_id}

- Modifies the slip with the given id
- If the id is not valid, then it will send a 400 status code back

- Set Header content-type to application/json
- The request body must be JSON and modeled:

```
{  
  "number": 1  
}
```

- If the request does not model the above, it will be rejected and send a 400 status code back
- If the 'number' is not unique, then it will ignore the request and send a 400 status code
- If you do not want to edit a specific item, then send its current value

PUT /slip/{slip_id}

- Puts the boat in this slip out to sea
- If the id is not valid, then it will send a 400 status code back
- Sets the current boat to be at sea
- Set the current slip to empty