

Conner Pappas, CS493: Assignment 5 - Build an Advanced Restful API

URL: <https://cpcs493advancedapi.appspot.com>

GET /ships

Description: Get all ships stored in the datastore. The request must include an 'Accept' header with either 'application/json' or 'text/html' as the value. The response is the collection of ships in the requested format with a 200 status code. Any other value for the 'Accept' header returns a 406 status code.

Example response:

Response code: 200

```
[
  {
    "name": "Test Ship 1",
    "type": "Ship 1",
    "length": 10,
    "id": "5636953047302144",
    "self": "https://cpcs493advancedapi.appspot.com/ships/5636953047302144"
  },
  {
    "name": "duudle",
    "type": "dyydle",
    "length": 69,
    "id": "5639445604728832",
    "self": "https://cpcs493advancedapi.appspot.com/ships/5639445604728832"
  },
  {
    "name": "deelde",
    "type": "doodle",
    "length": 45,
    "id": "5707702298738688",
    "self": "https://cpcs493advancedapi.appspot.com/ships/5707702298738688"
  }
]
```

GET /ships/{shipID}

Description: Get a response containing a one element array containing an object that contains the properties of the requested shipID in the URL. The request must include an 'Accept' header with either 'application/json' or 'text/html' as the value. The response is the information of the ship specified in the URL in the requested format with a 200 status code. Any other value for the 'Accept' header returns a 406 status code.

Example response:

```
Response code: 200
[
  {
    "name": "deelde",
    "type": "doodle",
    "length": 45,
    "id": "5707702298738688",
    "self": "https://cpcs493advancedapi.appspot.com/ships/5707702298738688"
  }
]
```

POST /ships

Description: Create a new ship containing the properties specified in the request body. The properties that can be specified by the user are the name of the ship, the type of ship, and the length of the ship. The request must contain a 'Content-Type' header with the value 'application/json' since the server only accepts json, else the server responds with a 415 status code. The response to a successful request is the id of the new ship with a 201 status code.

Example request:

```
{
  "name": "Test Ship 1",
  "type": "Ship 1",
  "length": 10
}
```

Example response:

```
Response code: 201
{"id": 5959639107633152}
```

PATCH /ships/{shipID}

Description: Modify a ship, specified in the URL, with the properties specified in the request body. The request must contain a 'Content-Type' header with the value 'application/json' since the server only accepts json, else the server responds with a 415 status code. The response to a successful request is a 303 status code, and a 'Location' header that contains a link to the updated ship.

Example request:

```
{
  "length": 42,
  "name": "Wiggly Woo Ha",
  "type": "Slippery ship"
}
```

Example response:

```
Response code: 303
Location -> https://cpcs493advancedapi.appspot.com/ships/5707702298738688
```

DELETE ships/{shipID}

Descriptions: Delete a ship, specified by shipID, from the datastore.

Example response:

```
Response code: 204
```

NOTE: Any attempts to send a DELETE or PATCH request to /ships results in a 405 status code