Student: Joaquin Saldana
REST Planning and Implementation

Google Cloud Link: https://rest-implementation-183107.appspot.com

Summary:

There are three types of models in the API and below are their properties:

1. Boat
    a. Name – string
    b. Type – string
    c. Length – integer
    d. At_sea – boolean
2. Departure History
    a. Departure date – string
    b. Departed date – string
3. Slip
    a. Slip number – integer
    b. Current boat – string
    c. Arrival date – string
    d. Departure history – structured property (of departure history) **EXTRA CREDIT**

Below are my URL's and the verbs (HTTP actions) for these URL's.

Boats

Create a Boat

POST /boats

Notes

- Boat entity has a property titled "at_sea" and automatically set to true when the boat is created. In order to change this property you will need to visit the **Mark a Boat At Sea** section
- If created successfully it will return a 201 response code

Name	Type	Description
name (required)	String	Name of the boat
type (required)	String	Type of boat

length (required)	Integer	Length of the boat
at_sea	Boolean	If vessel is at sea (auto default to true)

List All Boats

GET /boats

Notes:
- If successful, will return a 201 response code

List Specific Boat

GET /boats/{boat id}

Notes:
- If successful will return a 201 response code

Modify Boat

PATCH /boats/{boat id}

Notes:
- If successful, will return a 201 response code if not a 505 error code

Name	**Type**	**Description**
name (optional)	String	Name of the boat
type (optional)	String	Type of boat
length (optional)	Integer	Length of the boat

Modify Entire Boat

PUT /boats/{boat id}

Notes:
- If successful, will return a 201 response code if not a 505 error code

Name	**Type**	**Description**
name (required)	String	Name of the boat
type (required)	String	Type of boat
length (required)	Integer	Length of the boat

Delete Boat

DELETE /boats/{boat id}

Note:

- If boat is docked the slip will be updated accordingly and opened up and its' departure list is updated as well
- If successful, will return a 201 response code if not a 505 error code

Slips

Create A Slip

POST /slips

Notes:

- Slip number must not already exists, if not the entity will not be created and will throw a 505 error, else if successful will return a 201 response code

Name	**Type**	**Description**
slip_number (required)	Integer	Slip number

List All Slips

GET /slips

List Specific Slip

GET /slips/{slip id}

Modify A Slip

PATCH /slips/{slip id}

Notes:

- If slip number already exists it will not be updated and will throw a 505 error else will return a 201 response code

Name	**Type**	**Description**
slip_number (required)	Integer	Slip number

Delete Slip

DELETE /slips/{slip id}

Notes:

- If slip is occupied, the system will automatically change the boats at sea property to true
- If successful, will return a 201 response code if not a 505 error code

Docking Boat To Slip (similar to starring a Gist)

PUT /slips/{slip id}/dock/{boat id}

Notes:

- Will return 405 response code for invalid slip id
- Will return 404 response code for invalid boat id
- Will return 402 response code if boat is already docked
- Will return 403 response code if slip is already occupied

Name	**Type**	**Description**
arrival_date (required)	String	Date boat arrives to slip

Undock Boat From Slip (similar to starring a Gist)

PUT /boats/{boat id}/atsea/{slip id}

Notes:

- Will return 405 response code for invalid slip id
- Will return 404 response code for invalid boat id
- Will return 402 response code if boat is already at sea
- Will return 403 response code if slip is already empty

Name	**Type**	**Description**
departure_date (required)	String	Date boat departs to sea and leaves slip