

Website URL: <https://rest-assignment.appspot.com>

URLS:

1. /boats

- **GET /boats**– Returns all boats.
 - Data To Provide: None
- **POST /boats**– Creates a new boat.
 - Error Prevention: Input sent over by the client is checked to make sure valid input was entered for the boat's name, type, and length fields.
 - Data To Provide: JSON
 - JSON Valid Example:

```
{ "name": "Boat Name New",  
  "type": "Catamara",  
  "length": 28}
```
 - JSON Invalid (incorrect type) Example:

```
{"name": "Boat Name New",  
  "type": "Catamar",  
  "length": 28}
```

2. /boats/{boat_id}

Error Prevention For All Verbs Below– The {boat_id} specified in the URL is checked with each HTTP verb below to make sure that it exists in the cloud datastore. If the boat does not exist in the datastore, an error message is sent back to the client with a status of 400.

- **DELETE /boats/{boat_id}**– Deletes the boat specified by {boat_id} and empties the slip the boat was previously in.
 - Data To Provide: None
- **GET /boats/{boat_id}**– Returns a boat specified by {boat_id} along with a URL to view this boat.
 - Data To Provide: None
- **PATCH /boats/{boat_id}**– Sets a boat to be "at sea".
 - Error Prevention: If the boat specified by {boat_id} is already "at sea", then an error message is sent back with a status of 400.
 - Data To Provide: None
- **PUT /boats/{boat_id}**– Modifies an existing boat.
 - Error Prevention: Input sent over by the client is checked to make sure valid input was entered for the boat's name, type, and length fields.
 - Data To Provide: JSON
 - JSON Valid Example:

```
{ "name": "Modified Name",  
  "type": "Power Boat",  
  "length": 35}
```

- JSON Invalid (incorrect type) Example:

```
{ "name": "Modified Name",  
  "type": "Power Boats",  
  "length": 35}
```

- JSON Invalid (incorrect # of arguments sent) Example:

```
{"type": "Power Boats",  
  "length": 35}
```

3. /slips

- **GET /slips** – Returns all slips.
 - Data To Provide: None
- **POST /slips** – Creates a new slip
 - Error Prevention: Input sent over by the client is checked to make sure valid input was entered for the slip's number field and that it hasn't already been used by another slip.
 - Data To Provide: JSON
 - JSON Valid Example:

```
{ "number": 1}
```
 - JSON Invalid (Slip number already in use) Example:

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

4. /slips/{slip_id}/boat

Error Prevention For All Verbs Below– The {slip_id} specified in the URL is checked with each HTTP verb below to make sure that it exists in the cloud datastore. If the slip does not exist in the datastore, an error message is sent back to the client with a status of 400.

- **GET /slips/{slip_id}/boat**– Returns any boat current occupying any slip specified by {slip_id}.
 - Data To Provide: None
- **PUT /slips/{slip_id}/boat** – Assigns a boat a slip number specified by {slip_id}.
 - Error Prevention: The {boat_id} specified in the URL is checked to make sure that it exists in the cloud datastore. If the boat does not exist in the datastore, an error message is sent back to the client with a status of 400.

- Error Prevention: If the slip retrieved from the cloud datastore is already associated with a boat, then an error message is sent back with a status of 403.
- Error Prevention: If the boat retrieved from the cloud datastore is not at sea, then an error message is sent back with a status of 403.
- Data To Provide: JSON
- JSON Valid Example:

```
{ "current_boat": "{{my_boat}}",  
  "arrival_date": "7/14/2018"}
```
- JSON Invalid (Slip number already in use) Example:

```
{ "current_boat": "{{my_boat}}",  
  "arrival_date": "7/14/2018"}
```

5. /slips/{slip_id}

Error Prevention – The {slip_id} specified in the URL is checked with each HTTP verb below to make sure that it exists in the cloud datastore. If the slip does not exist in the database, an error message is sent back to the client with an status of 400.

- **DELETE /slips/{slip_id}** – Deletes the slip specified by {slip_id} and sets any boat it's associated with to "at_sea".
 - Data To Provide: None
- **GET /slips/{slip_id}**– Returns a slip specified by {slip_id}.
 - Data To Provide: None
- **PATCH /slips/{slip_id}**– Modifies a slip number specified by {slip_id}
 - Error Prevention: If the user sends more than 1 argument, then an error message is sent back with a status of 400.
 - Error Prevention: Input sent over by the client is checked to make sure valid input was entered for the slip's number field and that it hasn't already been used by another slip.
 - Data To Provide: JSON
 - JSON Valid Example:

```
{ "number": 1}
```
 - JSON Invalid (Slip number already in use) Example:

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