

Create a boat

POST /boats

Input

Name	Type	Description
name	string	Required. Name of boat.
type	string	Required. Type of boat.
length	integer	Required. Length of boat.
at_sea	boolean	True. Indicates whether the boat is out at sea or at slip.

Name, type and length are all required when creating a boat. By default, all boats start at sea, hence, at_sea is set to true. When a call is made to POST /boats, a new boat will be created and added to the data store. For example:

```
{
  "name": "Name of boat",
  "type": "Type of boat",
  "length": 20
}
```

This will create a boat with name, type and length specified in the request body, and set at_sea to default which is true.

Response

Status: 200 OK

```
{
  "name": "Name of boat",
  "type": "Type of boat",
  "length": 20,
  "at_sea": true,
  "id": "abc123",
  "self": "/boats/abc123"
}
```

If at_sea is provided in the request body and has a value of false, the system will not create a new boat because this violates the requirement that all boat starts at sea.

Delete all boats

DELETE /boats

Delete all boats entities from the data store. Before deleting a boat entity, if the boat is currently at slip, the slip will be emptied. In addition, the boat and its information is added to the boat's history list.

Response

Status: 200 OK

View all boats

GET /boats

Display a list of all boat entities currently in our data store.

Response

Status: 200 OK

```
[
  {
    "name": "Boat1",
    "type": "Type of boat",
    "length": 20,
    "at_sea": true,
    "id": "abc123",
    "self": "/boats/abc123"
  },
  {
    "name": "Boat2",
    "type": "Type of boat",
    "length": 20,
    "at_sea": false,
    "id": "abcd1234",
    "self": "/boats/abcd1234"
  },
  ...
]
```

View a boat

GET /boats/{boatID}

Display a boat entity.

Response

Status: 200 OK

```
{
  "name": "Boat1",
  "type": "Type of boat",
  "length": 20,
  "at_sea": true,
  "id": "abc123",
  "self": "/boats/abc123"
}
```

```
}
```

Delete a boat

DELETE /boats/{boatID}

Delete a boat entity. Before deleting, empty the slip the boat is currently occupying and add boat information to boat's history list.

Response

Status: 200 OK

Edit a boat

PUT /boats/{boatID}

Input

Name	Type	Description
name	string	New name of boat. (optional)
type	string	New type of boat. (optional)
length	integer	New length of boat. (optional)

Edit the information of a boat with new information sent from the request body. For example, we can change the name, type and length of a boat with the following in the request body:

```
{
  "name": "Modified name",
  "type": "Modified type",
  "length": 40
}
```

Response

Status: 200 OK

```
{
  "name": "Modified name",
  "type": "Modified type",
  "length": 40,
  "at_sea": true,
  "id": "abc123",
  "self": "/boats/abc123"
}
```

Setting a boat to at sea

PUT /boats/{boatID}

We can set the boat to at sea by setting the boat's `at_sea` to true. Before setting the boat to at sea, make sure to empty the slip the boat is currently occupying. For example, a call with the following body will set the boat from a slip to at sea:

```
{
  "at_sea": true
}
```

Response

Status: 200 OK

```
{
  "name": "Modified name",
  "type": "Modified type",
  "length": 40,
  "at_sea": true,
  "id": "abc123",
  "self": "/boats/abc123"
}
```

If a request is sent to set `at_sea` to false, the system will return an error as the only modification using this request is setting `at_sea` to true.

Replacing a boat

`PATCH /boats/{boatID}`

Replace an existing boat with a new boat, but using the same boat's id. All new boat should start at sea.

Input

Name	Type	Description
name	string	Required. Name of boat.
type	string	Required. Type of boat.
length	integer	Required. Length of boat.

For example:

```
{
  "name": "New name",
  "type": "New type",
  "length": 80
}
```

Response

Status: 200 OK

```
{
  "name": "New name",
```

```
    "type": "New type",
    "length": 80,
    "at_sea": true,
    "id": "abc123",
    "self": "/boats/abc123"
}
```

Create a new slip

POST /slips

Input

Name	Type	Description
number	integer	Required. A unique slip's number.

Create a new slip with information provided in request body. Check that slip number provided is unique by checking if number is already in used by other slip. If number is used by other slip, return with an error.

If slip number is unique, check if current_boat or arrival_date information is provided in request body. If either current_boat or arrival_date is provided, system will generate an error as all new slips should be empty.

Response

Status: 200 OK

```
{
  "number": 1,
  "current_boat": null,
  "arrival_date": null,
  "self": "/slips/xyz123",
  "id": "xyz123"
}
```

View all slips

GET /slips

Display a list of all slip entities.

Response

Status: 200 OK

```
[
  {
    "number": 1,
```

```

        "current_boat": "abc123",
        "current_boat_url": "/boats/abc123",
        "arrival_date": "2017-07-11",
        "self": "/slips/xyz123",
        "id": "xyz123"
    },
    {
        "number": 2,
        "current_boat": null,
        "arrival_date": null,
        "self": "/slips/xyz1234",
        "id": "xyz1234"
    },
    ...
]

```

Delete all slips

DELETE /slips

Delete all slip entities currently in data store. If a boat is currently occupying the slip we are deleting, set the boat to at sea before deleting the slip.

Response

Status: 200 OK

Get a single slip

GET /slips/{slipID}

Display a single slip entity.

Response

Status: 200 OK

```

{
    "number": 1,
    "current_boat": "abc123",
    "current_boat_url": "/boats/abc123",
    "arrival_date": "2017-07-11",
    "self": "/slips/xyz123",
    "id": "xyz123"
}

```

Delete a slip

DELETE /slips/{slipID}

Delete a single slip entity from data store. Before deleting, check if any boat is currently occupying the slip. If boat is occupying the slip, set boat to at sea.

Response

Status: 200 OK

Edit a slip number

PUT /slips/{slipID}

Edit information of a slip.

Input

Name	Type	Description
number	integer	A new unique slip's number. Optional

Check if new slip number is being used by another slip before editing the slip we are editing. If new slip number is used, return an error. If new slip number is not used, current slip number will be updated. For example:

```
PUT /slips/xyz1234
```

```
{
  "number": 10
}
```

Response

Status: 200 OK

```
{
  "number": 10,
  "current_boat": null,
  "arrival_date": null,
  "self": "/slips/xyz1234",
  "id": "xyz1234"
}
```

Arrival of boat to slip

PUT /slips/{slipID}

Input

Name	Type	Description
current_boat	string	A new boat's id.
arrival_date	string	A new arrival date.

First, check if current slip is being occupied by another boat. If current slip is occupied, return error message:

```
"Error 403 Forbidden Message"
```

If slip is empty (not occupied), assign boat to current slip, and update boat's at_sea status to false.

Response

Status: 200 OK

```
{
  "number": 10,
  "current_boat": "abc123",
  "current_boat_url": "/boats/abc123",
  "arrival_date": "2017-07-12",
  "self": "/slips/xyz1234",
  "id": "xyz1234"
}
```

Replace a slip

PATCH /slips/{slipID}

Input

Name	Type	Description
number	integer	Required. A unique slip's number.

Replace an existing slip with new slip but using the same slip's id. Check if new slip number is being used by another slip, return error message if slip number is used. Boat occupying the slip we are replacing should continue to occupy the same lot but with updated slip number. For example, if boat A is currently in slip 1, and we are replacing slip 1 with slip 2, then after the replacement, boat A should be in slip 2.

If new boat id or arrival_date is provided, return with error as a new slip should be either empty or occupied by the same boat.

```
PATCH /slips/xyz1234
{
  "number": 20
}
```

Response

Status: 200 OK

```
{
  "number": 20,
  "current_boat": "abc123",
  "current_boat_url": "/boats/abc123",
}
```



```
    "arrival_date": "2017-07-12",
    "self": "/slips/xyz1234",
    "id": "xyz1234"
  }
```

Get boat currently in slip

GET /slips/{slipID}/boats

Display information of current boat entity occupying this slip.

Response

Status: 200 OK

```
{
  "name": "Name of boat",
  "type": "Type of boat",
  "length": 20,
  "id": "abc123",
  "self": "/boats/abc123"
}
```

View all deleted boats

GET /boats/history

Display a list of all previously deleted boats.

Response

Status: 200 OK

```
[
  {
    "name": "Name of boat",
    "type": "Type of boat",
    "length": 20,
    "id": "abc123",
    "self": "/boats/abc123",
    "slip_number": 1
  },
  {
    "name": "Boat 2",
    "type": "Type 2",
    "length": 25,
    "id": "abc1234",
    "self": "/boats/abc1234",
    "slip_number": null
  },
  ...
]
```

Clear history of deleted boats

DELETE /boats/history

Delete all boats in history list.

Response

Status: 200 OK