INSTRUCTIONS AND SETUP FILE SUMMARY

Test results:	HarborMaster.postman_test_run.json
Collection of tests:	HarborMaster.postman_collection.json
Environment:	HarborMaster.postman_environment.json
Usage PDF:	API_Instructions.pdf (this file).
main.py	API code.
app.yaml	Configuration file.

Harbor Master API Instance: https://boatslipsbychris.appspot.com

Summary of API URL's for the below operations:

1. Create a new Boat: *	POST	/Boat
2. Delete a Boat:	DEL	/{{Boat_ID}}
3. Retrieve a Boat:	GET	/{{Boat_ID}}
4. Retrieve All Boats:	GET	/AllBoats
5. Create a new Slip: *	POST	/Slip
6. Delete a Slip:	DEL	/{{Slip_ID}}
7. Retrieve a Slip:	GET	/{{Slip_ID}}
8. Retrieve All Slips:	GET	/AllSlips
9. Dock a Boat to a Slip: *	PUT	/DockOps/{{Boat_ID}}
10. Cast off from a Slip: *	PATCH	/DockOps/{{Boat_ID}}
11. Modify a Boat's Attributes: *	PATCH	/{{Boat_ID}}
12. Replace a Boat's Attributes: *	PUT	/{{Boat_ID}}
13. Modify a Slip's Attributes: *	PATCH	/((Slip_ID}}
14. Replace a Slip's Attributes: *	PUT	/((Slip_ID}}

* Requires a payload constructed as a json object.

Detailed instructions on the Harbor Master API.

1. Create a new Boat: **POST** **/Boat**

POST payload:

```
{
        "name": "Boat Name 01",
        "type": "Boat Type 01",
        "length": 111
}
```

Returns a json object representing the Boat entity as follows:

```
{
 "at_sea": true,
 "length": 111,
 "type": "Boat Type 01",
 "name": "Boat Name 01",
 "id": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string"
}
```

Notice that the Boat entity id begins with 'Boat/'. Slip entity ID's are similarly constructed. This is to prevent you from doing odd things like attempting to dock a slip into another slip...

2. Delete a Boat: **DEL** **/{{Boat_ID}}**

Deletes the Boat entity specified by the "Boat_ID" string.
Providing an invalid Boat_ID or no Boat_ID will result in response status code 404.
Successful Boat entity deletion will return the html string: "*Boat has been deleted!*"
If the Boat is successfully deleted and was docked in a Slip, the affected slip's "current_boat" and "arrival_date" attributes will be updated to "null".

3. Retrieve a Boat: GET /{{Boat_ID}}

Returns a json object representing the Boat entity as follows:

```
{
 "at_sea": true,
 "length": 111,
 "type": "Boat Type 01",
 "name": "Boat Name 01",
 "id": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string"
}
```

4. Retrieve All Boats: GET /AllBoats

Returns an array of json objects representing the Boat entities as follows:

```
[
 {
  "at_sea": true,
  "length": 111,
  "type": "Boat Type 01",
  "id": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string"
  "name": "Boat Name 01"
 },
 {
  "at_sea": true,
  "length": 222,
  "type": "Boat Type 02",
  "id": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string",
  "name": "Boat Name 02"
 },
 {
  "at_sea": true,
  "length": 333,
  "type": "Boat Type 03",
  "id": "/Boat/ an_80_character_unique_alphanumeric_urlsafe_string",
  "name": "Boat Name 03"
 }
]
```

5. Create a new Slip: POST /Slip

POST payload: (empty payload)

```
{}
```

Returns: A json object representing the Slip. The slip number auto increments to the next available number starting at 1. Only deleting all the Slips will reset the slip counter to 1; redeploying, etc. will not!

```
{
  "arrival_date": null,
  "departure_history": {
    "departure_date": null,
    "departed_boat": null
  },
  "number": 1,
  "current_boat": null,
  "id": "/Slip/ an_80_character_unique_alphanumeric_urlsafe_string"
}
```

6. Delete a Slip: DEL /{{Slip_ID}}

On success, returns an html string: "Slip has been deleted!"
If there was a boat docked in the deleted slip, it's "at_sea" attribute will be automatically updated to "true".

7. Retrieve a Slip: GET /{{Slip_ID}}

Returns: A json object representing the Slip.

```
{
 "arrival_date": null,
 "departure_history": {
  "departure_date": null,
  "departed_boat": null
 },
 "number": 1,
 "current_boat": null,
 "id": "/Slip/an_80_character_unique_alphanumeric_urlsafe_string"
}
```

8. Retrieve All Slips: GET /AllSlips

Returns an array of json objects representing the Slip entities as follows: (The null fields will populate once you start docking and casting off your boats.)

```
[
 {
  "current_boat": null,
  "departure_history": {
      "departure_date": null,
      "departed_boat": null
  },
  "arrival_date": null,
  "id": "/Slip/an_80_character_unique_alphanumeric_urlsafe_string"
  "number": 1
 },
 {
  "current_boat": null,
  "departure_history": {
      "departure_date": null,
      "departed_boat": null
  },
  "arrival_date": null,
  "id": "/Slip/an_80_character_unique_alphanumeric_urlsafe_string",
  "number": 2
 }
]
```

9. Dock a Boat to a Slip: PUT /DockOps/{{Boat_ID}}

PUT payload:

```
{
        "number": 1,
            "arrival_date": "04/25/2017"
}
```

Note: The slip number is the slip you wish to dock at. Status Code 403 forbidden is returned if a boat is already in that slip number, otherwise, returns a json object representing the updated Slip entity as follows:

(The null fields will populate once you cast off your boat. On successful docking, the Boat that docked will automatically have its "at_sea" attribute updated to "false".

Returns: A json object representing the updated Slip attributes. Note, the json object depicted below represents the first time a boat has docked to a new slip. Subsequently casting off the Boat from this new slip will cause the departure_history, departure_date, and departed_boat attributes to be updated with this Boat, and then updated for all future Boats that reside in the slip when they cast off.

```
{
  "arrival_date": "04/25/2017",
  "departure_history": null,
  "departure_date": null,
  "number": 1,
  "departed_boat": null,
  "current_boat": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string",
  "id": "/Slip/an_80_character_unique_alphanumeric_urlsafe_string",
}
```

Attempting to dock a boat with an invalid Boat_ID (such as attempting to dock a slip into another slip) or a missing Boat_ID will result in 404 Not Found being returned.

10. Cast off from a Slip: PATCH /DockOps/{{Boat_ID}}

PATCH payload:

```
{
            "number": 1,
            "departure_date": "04/28/2017"
}
```

Returns: A json object representing the updated Slip attributes. Note the arrival_date, departure_date, departed_date, and current_boat attribute updates.

```
{
  "arrival_date": null,
  "departure_date": "04/28/2017",
  "number": 1,
  "departed_boat": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string",
  "current_boat": null,
  "id": "/Slip/an_80_character_unique_alphanumeric_urlsafe_string"
}
```

Note 1: Returns status code 400 Bad Data for attempting to cast off a boat that is not currently in the slip you specify in the payload.
Note 2: On successfully casting off, the Boat that cast off will automatically have its "at_sea" attribute updated to "true".

11. Modify a Boat's Attributes: PATCH /{{Boat_ID}}

PATCH Payload: (Only name, type, and length are allowed to modified. All other Boat attributes remain unchanged. Here we show just name and length being Patched, type remains unchanged.

```
{
        "name": "NewNameBoat 3",
        "length": 99
}
```

Returns: A json object representing the updated Boat attributes. Note type is not changed.

```
{
  "at_sea": false,
  "length": 99,
  "type": "Boat Type 03",
  "name": "NewNameBoat 3",
  "id": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string"
}
```

12. Replace a Boat's Attributes: PUT /{{Boat_ID}}

PUT Payload: Only attributes name, type, and length can be replaced. Omitting any of these attributes in the payload will result in the missing attribute being set to null. Demonstrated below is the case of attribute "length" being omitted from the payload.

```
{
   "type": "Assault Landing Craft 44",
    "name": "Predator44"
}
```

Returns a json object representing the Boat entity as follows:

```
{
  "at_sea": true,
  "length": null,
  "type": "Assault Landing Craft 44",
  "name": "Predator44",
  "id": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string"
}
```

13. Modify a Slip's Attributes: PATCH /((Slip_ID}}

PATCH Payload:

```
{
   "number": 5,
   "current_boat": "{{newBoat_1_ID}}",
   "arrival_date": "04/27/2017",
   "departed_boat": "{{newBoat_3_ID}}",
  "departure_date": "04/28/2017"
}
```

Note: the {{...}} are typical of Postman's environment variables. Substitute "/Boat/an_80_character_unique_alphanumeric_urlsafe_string" if not using Postman.

Returns: A json object representing the updated Slip attributes.

```
{
 "arrival_date": "04/27/2017",
 "departure_date": "04/28/2017",
 "number": 5,
 "departed_boat": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string",
 "current_boat": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string",
 "id": "/Slip/an_80_character_unique_alphanumeric_urlsafe_string"
}
```

NOTES:

Omitted payload attributes as noted above will not be modified.
Supplying a new slip number that is already taken will result in slip number being set to the next available slip number.
Attribute "current_boat" is only updated if the slip is empty. Use DockOps if this is the case.
If the slip was empty on a current_boat attribute specified in the payload, the affected Boat will automatically have its "at_sea" attribute updated to "false".

14. Replace a Slip's Attributes: PUT /((Slip_ID}}

PUT Payload:
Only attributes number, current_boat, arrival_date, departed_boat, and departure_date can be replaced.
<u>Omitting attributes current_boat, arrival_date, departed_boat, and departure_date in the payload will result in the missing attribute being set to null</u>.
If slip number is not included in the payload, it will not be changed.
If slip number is in the payload, and you attempt to change it to a slip number that is already taken, the next available slip number will be assigned.

```
{
   "number": 5,
   "current_boat": "{{newBoat_3_ID}}",
   "arrival_date": "04/27/2017",
   "departed_boat": "{{newBoat_2_ID}}",
   "departure_date": "04/26/2017"
}
```

Note: the {{...}} are typical of Postman's environment variables. Substitute "/Slip/an_80_character_unique_alphanumeric_urlsafe_string" if not using Postman.

Returns: A json object representing the updated Slip attributes.

```
{
  "arrival_date": "04/27/2017",
  "departure_date": "04/26/2017",
  "number": 6,
  "departed_boat": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string",
  "current_boat": "/Boat/an_80_character_unique_alphanumeric_urlsafe_string",
  "id": "/Slip/an_80_character_unique_alphanumeric_urlsafe_string"
}
```