REST Planning and Implementation

Class: CS 496
Name: Peter Moldenhauer
Date: 1/22/18

Live marina app URL (main page): https://myproject1-191416.appspot.com

URL's and verbs that work on the marina app:

POST /boats
- This creates a new boat and returns the newly created boat
- The body of the POST needs to be provided the following data and should take the form of: {"name":"str", "length":int, "type":"str"}
- The boat id will be automatically generated by the API and will be a string
- A newly created boat will start off "at sea"
- The newly created boat that is returned will contain "id", "name", "type", "self" (URL to itself), "length" and "at sea" fields
- Error checking: Will catch the error if the body of POST is in the wrong format and will then display the expected format.

GET /boats
- This will return a list of all the boats
- No data needs to be provided in body with GET
- Each boat in the list will contain "id", "name", "type", "self" (URL to itself), "length" and "at sea" fields

GET /boats/{boat_id}
- This will return the boat with the provided id (in the URL)
- The returned boat will contain "id", "name", "type", "self" (URL to itself), "length" and "at sea" fields
- No data needs to be provided in the body with GET
- Error checking: Will catch the error if the boat with the provided id does not exist (bad id in the URL)

PATCH /boats/{boat_id}
- This will update a single attribute of a given boat with the provided id (in the URL)
- The body of the PATCH needs to be provided the following data and should take the form of: {"name":"str"} or {"length":int} or {"type":"str"}
- Error checking: Will catch the error if the body is in the wrong format and will then display the expected format. Will also catch the error if the boat with the provided id does not exist (bad id in the URL).

PUT /boats/{boat_id}

- This will update all three attributes of the boat (name, length, type) with the provided id (in the URL)
- The body of the PUT needs to be provided the following data and should take the form of: {“name”:”str”, “length”:int, “type”:”str”}
- Error checking: Will catch the error if the body of PUT is in the wrong format and will then display the expected format. Will also catch the error if the boat with the provided id does not exist.

DELETE /boats/{boat_id}

- This will delete the boat with the provided id (if the boat is already “at sea”). If the boat with the provided id is currently in a slip, then this will set the slip to be empty and the boat will then be deleted.
- No data needs to be provided in the body with DELETE
- Error checking: Will catch the error if the boat with the provided id does not exist (bad id in the URL)

POST /slips

- This will create a new slip and returns the newly created slip
- The body of the POST needs to be provided the following data and should take the form of: {“number”: int}
- The slip id will be automatically generated by the API
- A newly created slip will have the “current_boat” and “arrival_date” fields initially set to be empty
- The newly created slip that is returned will contain “id”, “number”, “current_boat”, “self” (URL to itself) and “arrival_date” fields
- Error checking: Error checking: Will catch the error if the body of POST is in the wrong format and will then display the expected format. Will also catch the error if the provided slip number is already in use by another slip

GET /slips

- This will a list of all the slips
- No data needs to be provided in body with GET
- Each slip in the list will contain “id”, “number”, “current_boat”, “self” (URL to itself) and “arrival_date” fields

GET slips/{slip_id}

- This will return the slip with the provided id (in the URL)
- The returned slip will contain “id”, “number”, “current_boat”, “self” (URL to itself) and “arrival_date” fields
- No data needs to be provided in the body with GET
- Error checking: Will catch the error if the slip with the provided id does not exist (bad id in the URL)

PATCH /slips/{slip_id}

- This will update a single attribute of the slip with the provided id

- The body of the PATCH needs to be provided the following data and should take the form of: {“number”: int}
- Error checking: Will catch the error if the body is in the wrong format and will then display the expected format. Will also catch the error if the slip with the provided id does not exist (bad id in the URL). Will also catch the error if the selected slip number is already in use.

DELETE /slips/{slip_id}

- This will delete the slip with the provided id if the slip is currently empty. If the slip is not empty, this will first send the boat in the slip out to sea and then delete the slip.
- No data needs to be provided in the body with DELETE
- Error checking: Will catch the error if the slip with the provided id does not exist (bad id in the URL)

GET /marina/{slip_id}

- This will return the boat in the slip with the provided id in the URL – in other words, it will let you view if a boat is in a given slip based on the slip id in the URL.
- If the slip is not empty, it will return the boat with the containing “id”, “name”, “type”, “self” (URL to itself), “length” and “at sea” fields
- No data needs to be provided in the body with GET
- Error checking: Will catch the error if the slip with the provided id does not exist. (bad id in the URL)

PUT /marina/{slip_id}

- This will put the boat id (given in body of PUT) in the specific slip based on the slip id (provided in the URL)
- The body of the PUT needs to be provided the following data and should take the form of: {“current_boat”:”boat_id”, “arrival_date”:”YYY-MM-DD”}
- Error checking: Will catch the error if the body is wrong and will then display the expected format. Will also catch the error if the boat with the provided id does not exist. Will also catch the error if the slip with the provided id does not exist. Will also catch the error if the slip with the provided id already has another boat in it. Will also catch the error if the boat with the provide id is already in another slip.

DELETE /marina/{slip_id}

- This will send the boat out to sea that is in the given slip (slip id given in URL)
- No data needs to be provided in the body with DELETE
- Error checking: Will catch the error if the slip with the provided id does not exist (bad id in the URL). Will catch the error if the slip with the provided id does not have a boat in it.