

1. The link to your working REST service deployed on GCP. If the link you submit has "localhost" in it your project isn't on the cloud and will not receive any points.
<https://swimming-boat.uw.r.appspot.com>
2. A short (half to one page) critique of the following 3 endpoints (endpoint = HTTP method + URL)
 - i. Modify a boat
To modify a boat it will be to use:
HTTP method: PATCH
URL: https://swimming-boat.uw.r.appspot.com/boats/{{boat_id}}
Endpoint: PATCH https://swimming-boat.uw.r.appspot.com/boats/{{boat_id}}
To make changes to the boat variables name, type and length in the body put json:

```
{  
  
  "name": "Odyssey II",  
  "type": "Mega Yatch",  
  "length": 200  
  
}
```
 - ii. A boat arrives at a slip
To add a boat to the slip:
HTTP method: PUT
URL: https://swimming-boat.uw.r.appspot.com/slips/{{slip_id}}/{{boat_id}}
Endpoint: PUT https://swimming-boat.uw.r.appspot.com/slips/{{slip_id}}/{{boat_id}}
The code will first check if the client key has the slip id and information, if there are things in the client key then the program sets a variable slip to hold the information that the client has. After checking that the boat exists it will set in slips which boat is in the slip. After the slip is updated, then the boat will be updated with which slip it has been put in
 - iii. A boat departs from a slip
To remove a boat from the slip:
HTTP method: DELETE
URL: https://swimming-boat.uw.r.appspot.com/slips/{{slip_id}}/{{boat_id}}
Endpoint: DELETE https://swimming-boat.uw.r.appspot.com/slips/{{slip_id}}/{{boat_id}}
To remove a boat from the slip, the code will first check if there is a slip at the slip_id, if there it will check into the slip and check if the slip has any information, if there is then check if the variable current_boat has a boat, if there is a boat remove the boat and update the slip setting the current_boat to None, if there isn't then return an error.
- b. In your critique evaluate the design of these three endpoints.
 - i. Is the chosen HTTP method/verb appropriate for the functionality? Or do you think a different HTTP method is more appropriate?
For the Modify a boat we used Patch, I think Patch is appropriate in this case since Patch supplies a set of instructions to modify the boat. Put would be

another possible HTTP method since it takes the original version of the resource and replaces it. However, since we're only keeping boats with valid variables and information Patch is much better since we can specify what we want to happen and what should be done.

For the boat arrives at the slip we used Put, here I think That Patch could have been another way for this functionality to be used, it allows there to be more constraints on how to add boats to a slip, since sometimes some slips could hold more than one. But in this case there can only be one boat on the slip

For the final, boat departs the slip we use Delete which is appropriate from this functionality since we're removing the boat from the slip. We aren't trying to remove the boat or slip from the database, we just want to remove their connection to one another.

- ii. Does the URL pattern follows RESTful principles? If not, what would a better URL pattern?

The URL patterns follows the RESTful principles. When we call for slips or boats we get the text in the json format. Also in the postman we can see that the HTTP method is put at the front, then the url. And the result we get after running postman is a json output. Also, through the HTTP request, we can create, update, read and delete data. This assignment uses HTTP for all four CRUD (Create, Read, Update, Delete) operations