List of things to test against the API

1. Create a boat successfully
   1. Either check the variables are correct
   2. Check the response code is correct

Test Code

var data = JSON.parse(responseBody);

postman.setGlobalVariable("boat-link", data['self']);

postman.setGlobalVariable("boat-id", data['boat\_id']);

tests['Boat length'] = responseBody.has(25);

tests['Boat name'] = responseBody.has("Boat 7");

tests['Boat type'] = responseBody.has("Flat Boat");

tests["Successfully Created Boat (201)"] = responseCode.code === 201;

1. Create a slip successfully
   1. Either check the variables/properties are correct
   2. Check the response code is correct

var data = JSON.parse(responseBody);

postman.setGlobalVariable("slip-link", data['self']);

postman.setGlobalVariable("slip-id", data['slip\_id']);

tests['Slip Number'] = responseBody.has(5);

tests["Successfully Created Boat (201)"] = responseCode.code === 201;

1. Creating a slip with a slip number that already exists
   1. Check the returned response code is correct
   2. Check the message response is correct

tests["Attempting to create a boat w/ same slip number fails"] = responseCode.code === 505;

tests["Silps Exists Message"] = responseBody.has("Slip number already exists")

1. Create more boats and slips for testing
2. GET the entire list of boats to see if the GET boats w/ no boat id returns all of the boats
   1. Check the response code and see if it’s 201
   2. Check if the number of elements in the list is greater than 1 to make sure it retrieved all of them

function greaterThanOne(num)

{

if (num > 1)

{

return true;

}

return false;

}

var responseLength = JSON.parse(responseBody).length;

tests["GET request returned more than one boat"] = greaterThanOne(responseLength);

tests["Successfully Retrieved List of Boats (201)"] = responseCode.code === 201;

1. GET a boat
   1. Make sure the response code is correct
   2. Make sure the boat contents or ID is correct

tests["Boat ID is in response body"] = responseBody.has(postman.getGlobalVariable("boat-id"));

tests["Successfully Retrieved single boat (201)"] = responseCode.code === 201;

1. GET the entire list of slips to see if the GET slips w/ no slip id returns all of the slips
   1. Check the response code and see if its 201
   2. Check if the number of elements in the list is greater than 1 to make sure it retrieved all of them

function greaterThanOne(num)

{

if (num > 1)

{

return true;

}

return false;

}

var responseLength = JSON.parse(responseBody).length;

tests["GET request returned more than one slip"] = greaterThanOne(responseLength);

tests["Successfully Retrieved List of Slips (201)"] = responseCode.code === 200;

1. GET a slip
   1. Make sure the response code is correct
   2. Make sure the slip content or ID is correct

tests["Slip ID is in response body"] = responseBody.has(postman.getGlobalVariable("slip-id"));

tests["Successfully Retrieved single slip (201)"] = responseCode.code === 200;

1. PATCH / modify a boat
   1. Makes sure the response is ok 201
   2. Make sure the content of the boat modified is changed
   3. Make sure the ID is the same to ensure it’s the same boat

tests["Boat Type Was Changed"] = responseBody.has("Small Boat");

tests["Boat ID in response is same as passed in URL"] = responseBody.has(postman.getGlobalVariable("boat-id"));

tests["Successfully Retrieved single slip (201)"] = responseCode.code === 201;

1. PUT / modify/replace a boat
   1. Make sure the response is ok
   2. Make sure the content of the boat modified is changed
   3. Makes sure the ID is the same to ensure it’s the same boat

tests["Boat Type Was Changed"] = responseBody.has("Mini Boat");

tests["Boat Length Was Changed"] = responseBody.has(26);

tests["Boat Name Was Changed"] = responseBody.has("Boat 17");

tests["Boat ID in response is same as passed in URL"] = responseBody.has(postman.getGlobalVariable("boat-id"));

tests["Successfully Retrieved single slip (201)"] = responseCode.code === 201;

1. PATCH modify a slip number
   1. Make sure the response is ok
   2. Make sure the property / slip\_number is changed

tests["Slip Number Was Changed"] = responseBody.has(6);

tests["Slip ID in response is same as passed in URL"] = responseBody.has(postman.getGlobalVariable("slip-id"));

tests["Successfully modified single slip (201)"] = responseCode.code === 201;

1. PATCH modify a slip number with a number that already exists
   1. Make sure the response code is an error
   2. Make sure the message says that the

tests["Silps Exists Message"] = responseBody.has("Slip number already exists")

tests["Successfully Failed to Modify Slip With Slip Number Already Present)"] = responseCode.code === 505;

1. PUT a few boats to a few slips (DOCK)
   1. Make sure the response code is ok
   2. Make sure the arrival date is updated
   3. Make sure the current boat is updated
   4. Make sure the list of departed dates is saved (of departed lists) so we can check it later
2. PUT a few boats to sea
   1. Make sure the response code is ok
   2. Make sure the arrival date is none
   3. Make sure the current boat is none
   4. Make sure the list of departed dates is incremented by one / or that it has the boat id in the list
3. DELETE a boat that is not docked
   1. Make sure the response code is ok
   2. Make sure the boat is no longer in the list of boats
      1. May need to do a get message here
4. DELETE a slip that is not occupied
   1. Make sure the response code is ok
   2. Make sure the slip is no longer in the list of slips
5. DELETE a boat that is docked
   1. Make sure the slip updates
   2. Make sure the boat is deleted w/ response code
   3. Make sure the slip shows no current boat
6. DELETE a slip that is occupied
   1. Make sure the boat is updated to show at sea
   2. Make sure the slip is deleted w/ response code