

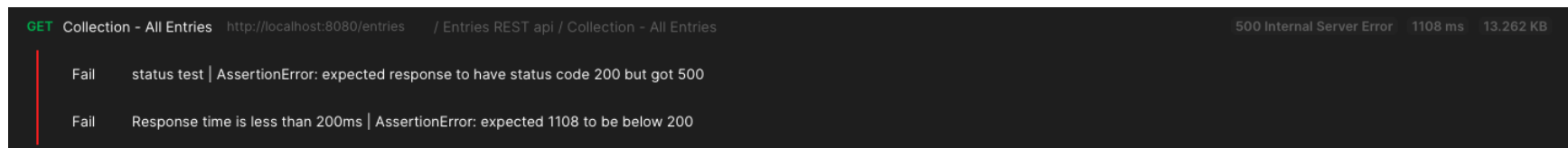
Postman Testing

Description:

Below are screenshots of Postman testing for the RESTful api service. For more detailed testing, please see [Postman Testing.Json](#) in GitHub.

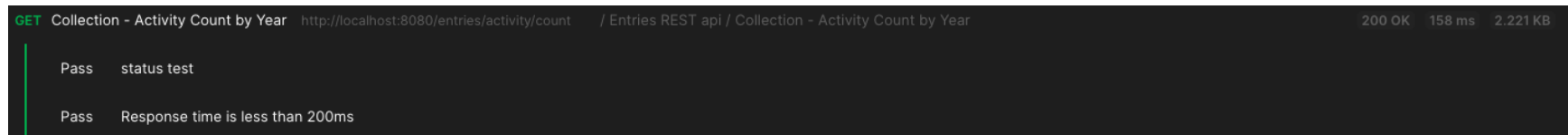
Analysis of Tests:

GET Collection - All Entries:



The test for this GET request failed. Reveals that further debugging and review is needed. However, it is not of concern as this request is not utilized in developing the front-end. Moreover, this reveals we may need to revisit EntryController, EntryService, or Entry classes.

GET Collection - Activity Count by Year:



Test passed - no concern.

GET Collection - Calories per Year:

GET	Collection - Calories per Year	http://localhost:8080/entries/activity/calories	/ Entries REST api / Collection - Calories per Year	200 OK	165 ms	2.397 KB
Pass	status test					
Pass	Response time is less than 200ms					

Test passed - no concern.

POST Add Entry/ies as JsonArray:

POST	Add Entry/ies as JsonArray	http://localhost:8080/entries/save/jsonarray	/ Entries REST api / Add Entry/ies as JsonArray	200 OK	13 ms	464 B
Pass	status test					
Pass	Response time is less than 200ms					

Test passed - no concern.

GET Collection - All Activities:

GET	Collection - All Activities	http://localhost:8080/activities	/ Activity REST api / Collection - All Activities	200 OK	7792 ms	6842.872 KB
Pass	status test					
Fail	Response time is less than 200ms AssertionError: expected 7792 to be below 200					
Fail	Test Length: AssertionError: expected 30752 to equal 30750					
Pass	Test Values:					

Response time is not ideal, however, it is important to note that this request gets all activities, thus it is expected that the response time will be longer. The threshold does not have to be below 200 ms, and could be increased. However, this does reveal that the duration is not ideal to get all activities. Moreover, the second test failed because the number of elements does not equal the expected number of elements. This is due to two(2) elements being added while testing. Thus, the failed test, which looks at length, should pass when changing the expected value after adding additional activities to the activities collection.

GET Collection - Filter by Activity:

GET Collection - Filter by Activity			http://localhost:8080/activities/cycling	/ Activity REST api / Collection - Filter by Activity	200 OK	535 ms	248.402 KB
Pass	status test						
Fail	Response time is less than 200ms AssertionError: expected 535 to be below 200						
Pass	Test Length:						
Pass	Test Values:						

Response time is okay. Test reveals that the response is 535 ms, which is fast in comparison to 7,792 in the previous test.

POST Add Activity:

POST Add Activity			http://localhost:8080/activities/add?date=20210818&activity=kayaking&startTime=0.0&endTime=0.0&duration=60&distance=5&steps=0&calories=200	/ Activity REST api / Add Activity	200 OK	86 ms	459 B
Pass	Successful POST Request:						
Pass	Response time is less than 200ms:						

Test passed - no concern.

PUT Update Activity:

The screenshot shows a REST client interface with a PUT request to `http://localhost:8080/activities/update/611de23ca557836537e198e5?date=20210818&activity=kayaking&startTime=0.0&endTime=0.0&duration=100&distance=1000&steps=0&calories=...`. The **Tests** tab is active, displaying two test scripts. The first test, "Successful PUT Request", passes. The second test, "Response time is less than 200ms", fails with an assertion error: "AssertionError: expected 219 to be below 200". The **Test Results** tab at the bottom shows the same results: one pass and one fail.

```
1 // tests for successful HTTP request
2 pm.test("Successful PUT Request: ", function () {
3     pm.response.to.have.status(200);
4     pm.response.to.be.ok;
5 })
6
7 pm.test("Response time is less than 200ms: ", function () {
8     pm.expect(pm.response.responseTime).to.be.below(200);
9 });
10
```

Test scripts are written in JavaScript, and are run after the response is received. [Learn more about tests scripts](#)

SNIPPETS

- [Get an environment variable](#)
- [Get a global variable](#)
- [Get a variable](#)
- [Get a collection variable](#)
- [Set an environment variable](#)
- [Set a global variable](#)

Status: 200 OK Time: 219 ms Size: 442 B [Save Response](#)

All Passed Skipped Failed

PASS Successful PUT Request:

FAIL Response time is less than 200ms: | AssertionError: expected 219 to be below 200

Response time difference is miniscule so there is no concern.

DEL Delete Activity:

The screenshot shows a REST client interface with a DELETE request to `http://localhost:8080/activities/delete/611de23ca557836537e198e5`. The **Tests** tab is active, displaying two test scripts. The first test checks for a successful delete request (status 204 and ok). The second test checks if the response time is less than 200ms. The **Body** tab is also visible, showing the response status as **204 No Content**, with a time of **131 ms** and size of **383 B**. The **Test Results** tab shows 1/2 results.

```
1 // tests for successful HTTP request
2 pm.test("Successful Delete Request: ", function () {
3     pm.response.to.have.status(204);
4     pm.response.to.be.ok;
5 })
6
7 pm.test("Response time is less than 200ms: ", function () {
8     pm.expect(pm.response.responseTime).to.be.below(200);
9 });
10
```

Test scripts are written in JavaScript, and are run after the response is received. [Learn more about tests scripts](#)

SNIPPETS

- [Get an environment variable](#)
- [Get a global variable](#)
- [Get a variable](#)
- [Get a collection variable](#)
- [Set an environment variable](#)
- [Set a global variable](#)

Status: 204 No Content Time: 131 ms Size: 383 B [Save Response](#)

Body Cookies Headers (12) Test Results (1/2)

Pretty Raw Preview Visualize Text

It appears the request is successful, however, the test reveals that the response time is slower than expected. This is interesting because it should not take a long time when given an id.

GET Collection - All Places:

GET	Collection - All Places	http://localhost:8080/places	/ Place REST api / Collection - All Places	200 OK	2103 ms	1843.664 KB
Pass	status test					
Fail	Response time is less than 200ms	AssertionError: expected 2103 to be below 200				
Fail	Test Length:	AssertionError: expected 11155 to equal 11154				

Response status passed, however, response time is not ideal. It was expected that this test would fail because this request is getting the entire places collection. Moreover, Test Length did not pass because a POST request was made when testing, thus, the assertion error was to be expected.

GET Collection - Filter by Places:

GET	Collection - Filter by Places	http://localhost:8080/places/place/Home	/ Place REST api / Collection - Filter by Places	200 OK	364 ms	182.476 KB
Pass	status test					
Fail	Response time is less than 200ms	AssertionError: expected 364 to be below 200				
Pass	Test Length:					
Pass	Test Values:					

Response status passed. Response time is 364 ms, which is a small increase from 200 ms. There's not too much to be concerned with.

GET Collection - Filter by Year:

GET	Collection - Filter by Year	http://localhost:8080/places/year/2016	/ Place REST api / Collection - Filter by Year	200 OK	3042 ms	418.365 KB
Pass	Status Test					
Fail	Response time is less than 200ms:	AssertionError: expected 3042 to be below 200				
Pass	Test Length:					
Pass	Test Values:					

Response time is not ideal, however, this can be explained because this request is searching the collection for all places that were visited in the given year. This can be marked for further review to make this request faster.

GET Collection - Place Count:

GET	Collection - Place Count	http://localhost:8080/places/count	/ Place REST api / Collection - Place Count	200 OK	2442 ms	2.856 KB
Pass	status test					
Fail	Response time is less than 200ms:	AssertionError: expected 2442 to be below 200				

Response time is to be expected because this request is creating a map using a query to get count of the number of times the place appears in the places collection.

POST Add Place:

POST	Add Place	http://localhost:8080/places/add?date=20210821&place=The Container Store&type=retail&latitude=-14.5&longitude=56.7&fourSquareId=1234	/ Place REST api / Add Place	200 OK	102 ms	459 B
Pass	Successful POST Request:					
Pass	Response time is less than 200ms:					

Tests passed - no concern.

PUT Update Place:

PUT <http://localhost:8080/places/update/611de29fa557836537e198e6?date=20130301&place=Northeastern University&type=School&latitude=-45.11&longitude=121.05&fourSquareId=1234> Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> date	20130301	
<input checked="" type="checkbox"/> place	Northeastern University	
<input checked="" type="checkbox"/> type	School	
<input checked="" type="checkbox"/> latitude	-45.11	
<input checked="" type="checkbox"/> longitude	121.05	
<input checked="" type="checkbox"/> fourSquareId	1234	
Key	Value	Description

Body Cookies Headers (14) Test Results (1/2) Status: 200 OK Time: 208 ms Size: 458 B Save Response

All Passed Skipped Failed

PASS status test

FAIL Response time is less than 200ms | AssertionError: expected 208 to be below 200

Test failed due to response time, however, it differs by 8 ms. Thus it is not concerning as it is still a fast response.

DEL Delete Place:

The screenshot shows the Postman interface for a DELETE request. The URL bar at the top shows `http://localhost:8080/places/delete/611de29fa557836537e198e6` with a 'Send' button to the right. Below the URL bar are tabs for Params, Authorization, Headers (6), Body, Pre-request Script, Tests, and Settings. The 'Tests' tab is selected, displaying a JavaScript test script:

```
1 // tests for successful HTTP request
2 pm.test("status test", function () {
3     pm.response.to.have.status(204);
4 })
5 // very fast to delete
6 pm.test("Response time is less than 200ms", function () {
7     pm.expect(pm.response.responseTime).to.be.below(200);
8 });
9
10
```

On the right side of the Tests tab, there is a sidebar with information about test scripts and snippets. It includes a note: "Test scripts are written in JavaScript, and are run after the response is received. [Learn more about tests scripts](#)". Below this, under the heading "SNIPPETS", there are several links: "Get an environment variable", "Get a global variable", "Get a variable", "Get a collection variable", "Set an environment variable", and "Set a global variable".

Below the Tests tab, there are tabs for Body, Cookies, Headers (12), and Test Results (2/2). The 'Test Results' tab is selected, showing the results of the tests. At the top right of this section, it displays "Status: 204 No Content", "Time: 86 ms", "Size: 383 B", and a "Save Response" button. Below this, there are filter buttons: "All", "Passed", "Skipped", and "Failed". The "Passed" filter is selected, showing two test results:

- PASS** status test
- PASS** Response time is less than 200ms

Tests passed - no concern.