Moshiur Howlader

|  |  |
| --- | --- |
| Test Charter #1 – Explore Basic REST API Routes | |
| #AREAS  In this 60 min session, explore the basic REST API routes without authentication.  Postman Version 7.24.0 (7.24.0) used on MacOS 10.15.4 (Catalina). | |
| START  20/12/05 8:36 PM | |
| TESTER  Moshiur Howlader | |
| TASK BREAKDOWN  #SESSION SETUP: 20  #TEST DESIGN AND EXECUTION: 40  #BUG INVESTIGATION AND REPORTING: 40  #CHARTER VS. OPPORTUNITY: 90:10 | |
| DATA FILES  Refer to folder in project folder called test\_charter\_1. | |
| TEST NOTES  Manually verified all 15 REST API methods in Postman (GET, POST, PUT, PATCH, DELETE, COPY, HEAD, OPTIONS, LINK, UNLINK, PURGE, LOCK, UNLOCK, PROPFIND, VIEW). Other than what is mentioned below, all non-supported methods return an expect 405 METHOD NOT ALLOWED status. | |
| BUGS | 1. All 6 non-default OPTIONS endpoints return 200 OK status while not authenticated 2. All 6 non-default HEAD endpoints return 200 OK status while not authenticated |
| ISSUES | 1. For the HEAD command, the 401 UNAUTHORIZED endpoints is returned but no output message is given. |

|  |  |
| --- | --- |
| Test Charter #2 – Exercise Various Stack Operation REST API Routes | |
| #AREAS  In this 30 min session, explore the various stack operations while authenticated. Make sure the basic happy and exception paths work.  Postman Version 7.24.0 (7.24.0) used on MacOS 10.15.4 (Catalina). | |
| START  20/12/05 11:45 PM | |
| TESTER  Moshiur Howlader | |
| TASK BREAKDOWN  #SESSION SETUP: 35  #TEST DESIGN AND EXECUTION: 35  #BUG INVESTIGATION AND REPORTING: 30  #CHARTER VS. OPPORTUNITY: 90:10 (checked case when running Postman queries against disconnected server) | |
| DATA FILES  Refer to folder in project folder called test\_charter\_2. | |
| TEST NOTES  Basic exercising of core backend logic of the stack-ws REST api.  Test Case 1: Insert various items to empty stack.  Test Case 2: Attempt a POP operation on empty stack.  Test Case 3: Peek top most element of non-empty stack.  Test Case 4: Peek on empty stack.  Test Case 5: Apply Size and Clear command on a stack instance object.  When exceptions occurred in TC2 and TC4, status code 500 returned correctly. | |
| BUGS | None Found |
| ISSUES | None Found |

|  |
| --- |
| Test Charter #3 – Exercise Stack Operation REST API Routes via White Box Testing |
| #AREAS  In this 30 min session, explore the various stack operations while authenticated. After reviewing source code, attempt to come up with additional test cases compared to before. Due the simplicity of the source code, creating additional unique test cases will be challenging. Will exercise some API methods previously not exercised.  Postman Version 7.24.0 (7.24.0) used on MacOS 10.15.4 (Catalina). |
| START  20/13/05 1:15 AM |
| TESTER  Moshiur Howlader |
| TASK BREAKDOWN  #SESSION SETUP: 35  #TEST DESIGN AND EXECUTION: 35  #BUG INVESTIGATION AND REPORTING: 30  #CHARTER VS. OPPORTUNITY: 70:30 |
| DATA FILES  Refer to folder in project folder called test\_charter\_3. |
| TEST NOTES  Test Case 1: Halfway into stack operations, De-authenticate, and reauthenticate to check same stack contents exists.  Test Case 2: Attempt a PUT operation on empty stack.  Test Case 3: Attempt indexing a negative integer index.  Test Case 4: Attempt indexing with 0.0  Test Case 5: Exercise various URL endpoints with HEAD method.  Test Case 6: Exercise various URL endpoints with OPTIONS method.  Test Case 7: POST 9000 new Stacks to collection (999999999999999999 will take 1 Hr+)  For Test Case 7, might be worthwhile to investigate the maximum POST threshold to set backend code for to prevent users from deliberately crashing the server. |
| BUGS  For Test Case 5, HEAD method does not seem have a View function implementation for Authenticated users for valid method signatures. Please refer to Step 6 for terminal output error. Please refer to test\_charter\_3/tc\_HEAD\_method for steps to reproduce error. |
| ISSUES  For Test Case 3, perhaps have a different status code or error message indicating negative integers are not valid.  For Test Case 4, perhaps have some SLUG URL or routing logic to disregard the point zero in the number and automatically reroute the user to the appropriate stack index. |