

Postman API TEST based on user story requirements

Date: May.31.2023

Tester: Tester YJ

This report has been designed with the sole purpose of demonstrating my Postman API testing skills and is added to my GitHub Portfolio.

The main objective of this testing exercise is to validate the functionality of an API endpoint based on predefined user story requirements. The test cases have been meticulously designed to cover different scenarios and acceptance criteria, aiming to ensure a seamless increase in play count for registered users.

User Story:

As a registered user I should be able to increase play count.

Acceptance criteria:

1. The POST api/interaction/play will be developed.
2. Authorization is required (Bearer token).
3. Endpoint accepts following payload in Song field (example): {songID}
4. Song ID is required.
5. Verification:
 - Error message - Requested song doesn't exist.
 - Song Id does not exist - 404 Not Found
 - Success - 200 Ok

Test Case: Steps and Results

Test Case	Verify	Endpoint	Authorization	Payload	Expected Output	Actual Output	Pass /Fail / Blocked
KL 1	Verify the endpoint (POST api/interaction/play) is well developed and functional	POST api/interaction/play	Bearer token	{ "songID": "157a67deb" }	200 OK Play count should be increased in the postman response body and actual database storage (MySQL)	200 OK Play count is increased in the both postman response body and actual database storage (MySQL)	PASS
KL 2	Verify incorrect request method returns proper error message and status code	GET api/interaction/play	Bearer token	{ "songID": "157a67deb" }	405 Method Not Allowed Response message: "The GET method is not supported for this route. Supported methods: POST."	405 Method Not Allowed Response message: "The GET method is not supported for this route. Supported methods: POST."	PASS
KL 3	Verify a valid Authorization token is required for a successful request	POST api/interaction/play	Bearer valid_token	{ "songID": "157a67deb" }	200 OK Play Count should be increased	200 OK Play Count is increased	PASS
KL 4	Verify missing Authorization token returns proper error message and status code	POST api/interaction/play	No token provided	{ "songID": "157a67deb" }	401 Unauthorized Response message: "Authorization required. Please provide a valid Bearer token."	401 Unauthorized Response message: "Authorization required. Please provide a valid Bearer token."	PASS
KL 5	Verify invalid Authorization token returns proper error message and status code	POST api/interaction/play	Bearer invalid_token	{ "songID": "157a67deb" }	401 Unauthorized Response message: "Authorization required. Please provide a valid Bearer token."	401 Unauthorized Response message: "Authorization required. Please provide a valid Bearer token."	PASS
KL 6	Verify correct payload returns status code 200 OK and 'Increased Play Count'	POST api/interaction/play	Bearer token	{ "songID": "157a67deb" }	200 OK Response message: "Increased Play Count"	200 OK Response message: "Increased Play Count"	PASS
KL 7	Verify missing SONG ID in payload returns	POST api/interaction/play	Bearer token	{ }	422 Unprocessable Content	422 Unprocessable Content	PASS

	proper error message and status code				Response message: "The given data was invalid."	Response message: "The given data was invalid."	
KL 8	Verify "404" status code when the requested song doesn't exist	POST api/interaction/play	Bearer token	{ "songID": "7p42c5" }	404 Not Found	500 Internal Server Error	FAIL
KL 9	Verify error message "Requested song doesn't exist" when the requested song doesn't exist	POST api/interaction/play	Bearer token	{ "songID": "7p42c5" }	Response message: "Requested song doesn't exist."	Since KL8 has failed, the verification of KL9 is not confirmed at this time.	BLOCKED
KL 10	Verify error message and status code with an incorrect format of payload	POST api/interaction/play	Bearer token	{ "id": "157a67deb" " }	400 Bad Request Response message: "Invalid payload format. Please provide a payload with the 'songID' field."	400 Bad Request Response message: "Invalid payload format. Please provide a payload with the 'songID' field."	PASS

Fail and Blocked

KL8: This test case aims to verify the "404" status code when the requested song doesn't exist. However, the actual response returns a "500 Internal Server Error" instead of the expected "404 Not Found" status code. The test fails. A bug report has been generated and is attached to this report.

KL9: As KL8 fails to meet the expected result, this test case, which verifies the error message for a non-existent song, is currently blocked. The verification of KL9 is pending resolution of the issue identified in KL8.

A Screenshot of the Bug Report

DashboardsvProjects vIssues vBoards vTests vCreate

board

Backlog

Active sprints

Releases

Reports

Issues

Components

Tests

Test Automation

YJ | API: Increase Play Count | Incorrect payload(Song ID) does not return '404 Not Found' status code

Edit

Comment

Assign

More

To Do

Done

Workflow

Details

Type: Bug

Priority: Medium

Status: TO DO (View Workflow)

Resolution: Unresolved

Labels: IncorrectSongID, IncreasePlayCount, KL_API_TEST, Sprint8, Postman_API,

Description

Steps to reproduce:

1. Select the POST method and Request the URI ({{baseUrl}}api/interaction/play)

2. Select the Authorization as 'Bearer Token'

3. Click on 'Body' Tab and Type the value (Key: "song", Value: "7p42c5")

4. Click on 'Send' button

Expected result:

Code Status '404 Not found' should be returned with 'Incorrect Song ID'

Actual result:

Code Status: '500 Internal Server Error' is returned

Attachments

People

Assignee:

Reporter:

Votes:

Watchers:

Dates

Created:

Updated:

Agile

View on Board

The attached Image file in the Bug Report

The screenshot shows the Postman interface with a workspace named 'My Workspace'. The left sidebar shows a collection of API requests under 'YJ Requests' > '3. Song Interactions'. The selected request is 'POST Increase play count'. The request details show a POST method to the endpoint `{{baseUrl}}/api/interaction/play` with the 'Body' tab selected. The body is a form-data type with a single parameter: 'song' with the value 'Doesn't exist Song ID'. A yellow box highlights this value, and a yellow arrow points from it to the response body. The response status is 'Status: 500 Internal Server Error' with a time of 182 ms and a size of 15.23 KB. The response body is in JSON format, showing a detailed error message: 'SQLSTATE[23000]: Integrity constraint violation: 1452 Cannot add or update a child row: a foreign key constraint fails ('dbkoel`.`interactions`, CONSTRAINT 'interactions_song_id_foreign' FOREIGN KEY ('song_id') REFERENCES 'songs' ('id') ON DELETE CASCADE (SQL: insert into `interactions` (`song_id`, `user_id`, `updated_at`, `created_at`) values (7p42c5, 28711, 2823-05-31 19:51:48, 2823-05-31 19:51:48))'. The exception is 'Illuminate\\Database\\QueryException'. The response body also shows a file path disclosure: 'file': 'Disclosure due to privacy and security'.

POST `{{baseUrl}}/api/interaction/play` Send

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description
<input checked="" type="checkbox"/> song	Doesn't exist Song ID → 7p42c5	
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 500 Internal Server Error Time: 182 ms Size: 15.23 KB Save as Example

Pretty Raw Preview Visualize JSON

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
{
  "message": "SQLSTATE[23000]: Integrity constraint violation: 1452 Cannot add or update a child row: a foreign key constraint fails ('dbkoel`.`interactions`, CONSTRAINT 'interactions_song_id_foreign' FOREIGN KEY ('song_id') REFERENCES 'songs' ('id') ON DELETE CASCADE (SQL: insert into `interactions` (`song_id`, `user_id`, `updated_at`, `created_at`) values (7p42c5, 28711, 2823-05-31 19:51:48, 2823-05-31 19:51:48))",
  "exception": "Illuminate\\Database\\QueryException",
  "file": "Disclosure due to privacy and security",
  "line": 712,
  "trace": [
    {
      "file": "Disclosure due to privacy and security",
      "line": 672,
      "function": "runQueryCallback",
      "args": [
        "SELECT * FROM `songs` WHERE `id` = ?"
      ]
    }
  ]
}
```