

CHARTER

Identify capabilities and areas of potential instability of the “rest api todo list manager”.

Identify documented and undocumented “rest api todo list manager” capabilities.

For each capability, create a script or small program to demonstrate the capability.

Exercise each capability identified with data typical to the intended use of the application.

BUILD

runTodoManagerRestAPI-1.5.5.jar

AREA

Exploratory testing for API endpoints

ENVIRONMENT

macOS Monterey

START

9:00 PM 2/4/26

TESTER

Rachel Zeng 261139258 yuqi.zeng@mail.mcgill.ca

TIME

45min

SESSION NOTES

Without specific values:

9:01 PM (GET /categories) Return all the instances of category.

9:01 PM (HEAD /categories) Return 200 OK

9:02 PM (POST /categories) Return 400 Bad Request

9:03 PM (GET /categories) Return 405 Method Not Allowed

9:05 PM (DELETE /categories) Return 405 Method Not Allowed

9:06 PM (OPTIONS /categories) Return 200 OK

9:07 PM (PATCH /categories) Return 405 Method Not Allowed

9:08 PM (GET /categories/:id) Return 404 Not Found

9:09 PM (PUT /categories/:id) Return 404 Not Found

9:10 PM (POST /categories/:id) Return 404 Not Found

9:10 PM (DELETE /categories/:id) Return 404 Not Found

9:11 PM (OPTIONS /categories/:id) Return 200 OK
9:11 PM (HEAD /categories/:id) Return 404 Not Found
9:14 PM (PATCH /categories/:id) Return 405 Method Not Allowed
9:15 PM (GET /categories/:id/projects) Return 200 OK #BUG 1
9:17 PM (PUT /categories/:id/projects) Return 405 Method Not Allowed
9:17 PM (POST /categories/:id/projects) id_value=1, return 201 Created
9:19 PM (DELETE /categories/:id/projects) Return 405 Method Not Allowed
9:20 PM (OPTIONS /categories/:id/projects) Return 200 OK
9:21 PM (HAED /categories/:id/projects) Return 200 OK #BUG 2
9:22 PM (PATCH /categories/:id/projects) Return 405 Method Not Allowed
9:22 PM (GET /categories/:id/projects/:id) Return 404 Not Found, but expect 405 Method Not Allowed
9:23 PM (PUT /categories/:id/projects/:id) Return 405 Method Not Allowed
9:23 PM (POST /categories/:id/projects/:id) Return 404 Not Found, but expect 405 Method Not Allowed
9:25 PM (DELETE /categories/:id/projects/:id) Return 404 Not Found when id have no value or 1
9:27 PM (OPTIONS /categories/:id/projects/:id) Return 200 OK
9:28 PM (HEAD /categories/:id/projects/:id) Return 404 Not Found, but expect 405 Method Not Allowed
9:30 PM (PATCH /categories/:id/projects/:id) Return 405 Method Not Allowed
9:31 PM (GET /categories/:id/todos) Return 200 OK #BUG 3
9:32 PM (PUT /categories/:id/todos) Return 405 Method Not Allowed
9:33 PM (POST /categories/:id/todos) Return 400 Bad Request when only input id value
9:33 PM (DELETE /categories/:id/todos) Return 405 Method Not Allowed
9:35 PM (OPTIONS /categories/:id/todos) Return 200 OK
9:37 PM (HEAD /categories/:id/todos) Return 200 OK #BUG 4
9:37 PM (PATCH /categories/:id/todos) Return 405 Method Not Allowed
9:38 PM (GET /categories/:id/todos/:id) Return 404 Not Found, but expect 405 Method Not Allowed
9:39 PM (PUT /categories/:id/todos/:id) Return 405 Method Not Allowed
9:40 PM (POST /categories/:id/todos/:id) Return 404 Not Found, but expect 405 Method Not Allowed
9:40 PM (DELETE /categories/:id/todos/:id) Return 404 Not Found
9:43 PM (OPTIONS /categories/:id/todos/:id) Return 200 OK
9:43 PM (HEAD /categories/:id/todos/:id) Return 404 Not Found, but expect 405 Method Not Allowed
9:45 PM (PATCH /categories/:id/todos/:id) Return 405 Method Not Allowed

BUGS

1. GET /categories/:id/projects

When insert value that does not exist in project, expect error but get 200 OK.

The screenshot shows the Postman interface with a GET request to `http://localhost:4567/categories/:id/projects`. The 'Params' tab is selected, showing a 'Query Params' table with a single entry: 'Key' (Value) and 'Value' (Description). The 'Path Variables' table shows 'id' set to '4'. The response status is 200 OK, with a response time of 3 ms and a body size of 237 B. The response JSON is:

```
1 {
2   "projects": [
3     {
4       "id": "2",
5       "title": "",
6       "completed": "false",
7       "active": "false",
8       "description": ""
9     }
10   ]
11 }
```

2. HEAD /categories/:id/projects

When insert value that does not exist in project, expect error but get 200 OK.

The screenshot shows the Postman interface with a HEAD request to `http://localhost:4567/categories/:id/projects`. The 'Params' tab is selected, showing a 'Query Params' table with a single entry: 'Key' (Value) and 'Value' (Description). The 'Path Variables' table shows 'id' set to '4'. The response status is 200 OK, with a response time of 29 ms and a body size of 147 B. The response JSON is:

```
1
```

3. GET /categories/:id/todos

When insert value that does not exist in project, expect error but get 200 OK.

The screenshot shows the Postman interface with a GET request to `http://localhost:4567/categories/:id/todos`. The 'Params' tab is selected, showing a table with one row: Key 'Key' and Value 'Value'. The 'Path Variables' section shows a table with one row: Key 'id' and Value '4'. The response section shows a status of 200 OK with a response body of `{ "todos": [] }`.

4. HEAD /categories/:id/todos

When insert value that does not exist in project, expect error but get 200 OK.

The screenshot shows the Postman interface with a HEAD request to `http://localhost:4567/categories/:id/todos`. The 'Params' tab is selected, showing a table with one row: Key 'Key' and Value 'Value'. The 'Path Variables' section shows a table with one row: Key 'id' and Value '4'. The response section shows a status of 200 OK with a response body of '1'.

CONCERNS

1. For BUG1&3, users are access to non-exist data, should be warned by error message.
2. For BUG2&4, the return type is mismatch with actual situation.