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

Typical Scenarios – main functions and capabilities of Categories

Exploratory testing for API endpoints

Environment

MacOS 26.3

Postman software

START

4:30 PM 13/02/2026

TESTER

Emile Labrunie	261097953	emile.labrunie@mail.mcgill.ca
Peter Zhang	261016377	yanzhe.zhang@mail.mcgill.ca

#DURATION

45 minutes

TEST NOTES

/categories:

[GET /categories]]
- returned all the instances as expected

[HEAD /categories]
- returned nothing as no header exists , as expected

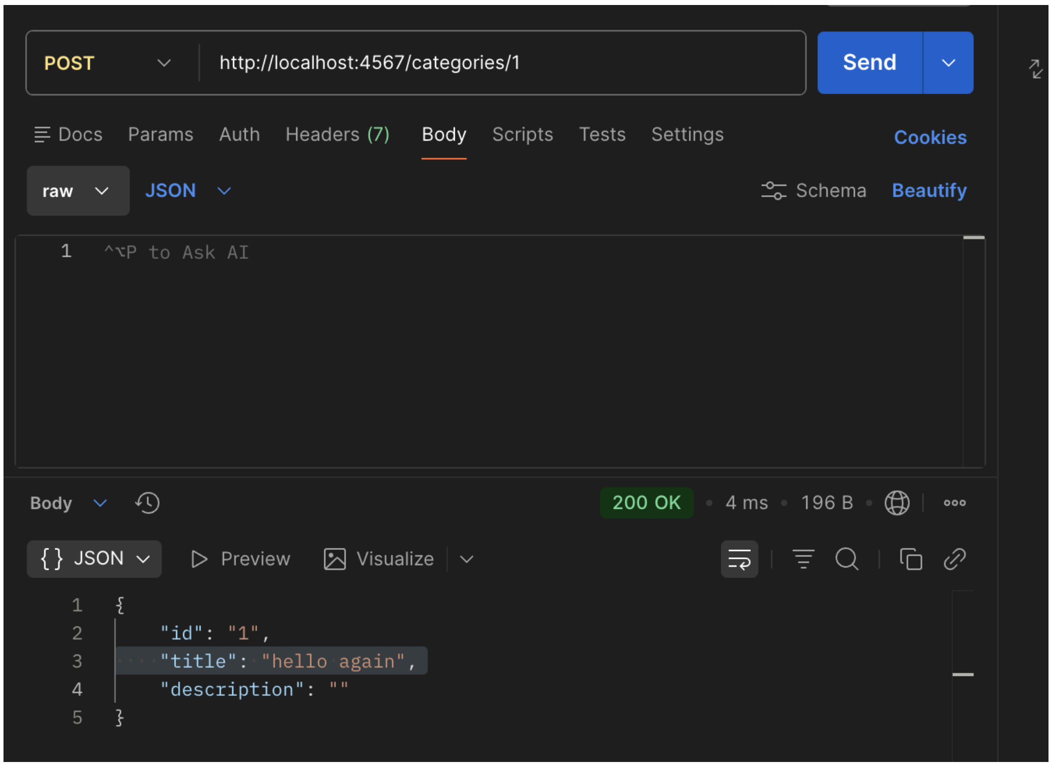
[POST /categories]
- in JSON: works as expected with the minimal required title , with valid payload
- in xml: works as expected with the minimal required title , with valid payload

/categories/:id

[GET /categories/:id]
- works with a valid one : gives iod title description of the aimed category
- error 404 not found when inputting a non existing id

[HEAD /categories/:id]
- for valid : found and empty as empty header
- fir invalid id : error 404 not found

[POST /categories/:id]
- in json : with existing id and valid payload , it successfully overwrites (amend)
- in json : invalid id error 404 not found
- in json : valid id , no body , RETURNED 200 OK AND NOTHING IS OVERWRITTEN, this is not expected behaviour , it should enforce a body and return an error



```
{
    "id": "1",
    "title": "hello again",
    "description": ""
}
```

- IN JSON : VALID ID, body with empty title , 400 bad request , title can not be spaces (works with 1 char)
- in XML : we get the same logic for everything tessted

[PUT /categories/:id]:
- In JSO : with existing id and valid payload , it successfully overwrites (amend)
- In json : with existing id butno body , it successfully rises an error and enforces a body.
- in XML : same testing , behaviorus are as expected

[DELETE /categories/:id]:
- with a valid it : deleted the instance as expected
- with a non existing id : rised an error as expected

EXTRA:

/docs:

- [GET /docs]: produces the html

/shutdown:

- [GET /shutdown]: shutdowned the server as expected
- [deleter/post/put/head + /shutdown] : gives 404 not found as expected

Areas covered:

endpoints related to project

** Ending session

Bugs logged

Areas of Potential Risk

[POST /categories/:id] is supposed to act as amend, and it currently doesn't enforce a request payload. This is not catastrophic because the entity's fields preserved in this scenario and not nulled. This still respects the basic constraints. Maybe this is as intended so we categorized it as Risk more than bug.

Capabilities

CRUD operations on Categories Entity in JSON

HEAD operation on Categories Entity in JSON

Refer to Capabilities Script: categoriesCapabilities.sh

Summary of session findings:

This session covered CRUD and HEAD operations for /Categories related endpoints, along with extra testing of the /docs and /shutdown endpoints. All core endpoints functioned as expected in both JSON and XML formats. No significant bugs were logged, though one area of potential risk was identified for amend behaviour.

POST /categories/:id does not enforce a request body, returning 200 OK with no changes when called with an empty body — mirroring the same risk found in Session 4 for the Todos entity. This pattern across both entities suggests a broader input validation gap in the amend endpoints.