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 ToDo

Exploratory testing for API endpoints

Environment

MacOS 26.3

Postman software

START

3:30 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

/todos

[GET /todos] Return all the instances of todo

- Correctly returned all the instances

[HEAD /todos] Headers for all the instances of todo

- Correctly returned all the headers

[POST /todos] We should be able to create todo without a ID using the field values in the body of the message

- Failed without a body, returned 400 error, as expected.
- Succeeded with a valid body

/todos/:id

[GET /todos/:id] return a specific instances of todo using a id

- When called with a valid id, Successfully returned the correct json object
- When called with an invalid id, Returned an error saying nothing found.

[HEAD /todos/:id] headers for a specific instances of todo using a id

- When provided a valid todo id, Returned the correct header,
- When trying to get header of a non existent todo, not found as expected

[POST /todos/:id] amend a specific instances of todo using a id with a body containing the fields to amend

- When the id don't exist, it said not found. This is expected for amend behaviour
- When id is valid, it amends as expected with a correct body.
- When id is valid and body is empty, it still returned a 200 status, but we think it is supposed to enforce a body.
- XML also works for both scenarios

[PUT /todos/:id] amend a specific instances of todo using a id with a body containing the fields to amend

- Invalid with no body
- Valid with body, and a valid id. as expected.
- Same for XML

[DELETE /todos/:id] delete a specific instances of todo using a id

- Deleted an existing instance of todo
- when deleted again, it said not found, which is expected from a deleted instance

/todos/:id/taskso; /todos/:id/tasksof/:id ; /todos/:id/categories ; /todos/:id/categories/:id

These **Relationship** todo endpoints were covered in session 1 / 2 .3

Areas covered:

---

endpoints related to Todo

Bugs logged

Areas of Potential Risk

[POST /todos/: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 Todos Entity in JSON

HEAD operation on Todos Entity in JSON

Refer to Capabilities Script: [todoCapabilities.sh]

Summary of session findings:

This session covered the full CRUD and HEAD operations for the Todos entity in JSON and XML formats. All core endpoints behaved as expected, with appropriate responses for valid and invalid inputs. No bugs were logged; relationship endpoints (tasksof, categories) were deferred as they were covered in Sessions 1–3.

One area of potential risk was noted: POST /todos/:id does not enforce a request body, returning 200 OK even when no body is provided. While non-destructive, this may indicate unintended permissiveness in the API's input validation.