**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**

Test “to do” feature

**ENVIRONMENT**

Windows 11

**START**

3:14PM 02/12/2025

**DURATION**

47 mins

**TESTER**

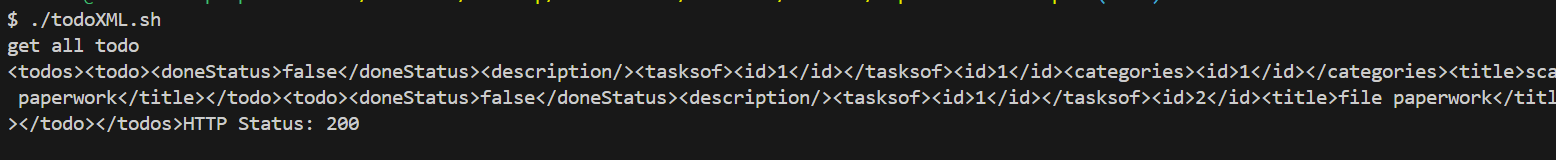
Neeshal Imrit 261068305 [neeshal.imrit@mail.mcgill.ca](mailto:neeshal.imrit@mail.mcgill.ca)

Dmytro Martyniuk 261118276 dmytro.martyniuk@mail.mcgill.ca

**SESSION NOTES**

2:55 PM – 12/02/2025: Set request headers to accept XML data.

3:00 PM: Sent a GET request to the /todos endpoint to retrieve all to-dos. Response returns a list containing two to-dos. The HTTP code was 200 which indicates a success.

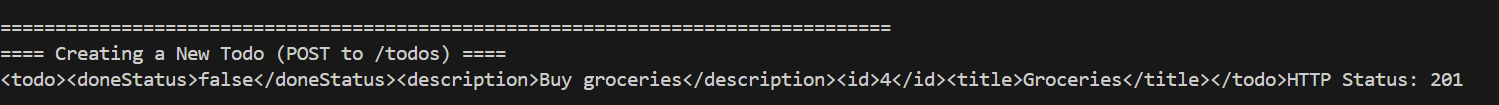


3:05 PM: Tested head requests to todos. Result was successful

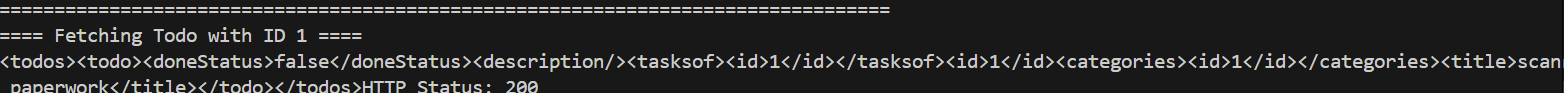
A screen shot of a computer

AI-generated content may be incorrect.

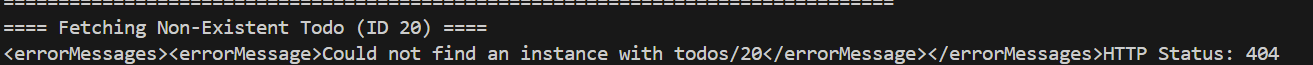
3:15 PM: Created a new todo with XML payload (POST /todos), specifying a title, doneStatus and description. The ID is auto-generated, and the TODO is successfully added.



3:19 PM: Sent a GET request for an existing category with ID = 1 (GET /todo/1). Returns the correct todo details.

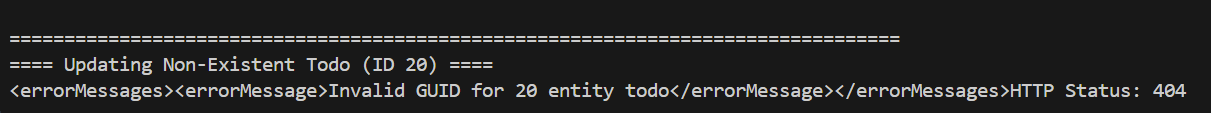


3:25 PM: Tested a GET request for a nonexistent todo (GET /todos/20). Returns a 404 error as expected.



3:30 PM: Sent a PUT request to update the doneStatus of Buy Groceries todo. Operation was successful with status code 201

3:32 PM: Attempted to update a nonexistent todo (PUT /todos/20). Returns a 404 error as expected.



3:35 PM: Sent a DELETE request to remove a todo with ID = 2 (DELETE /todos/2). Successfully deleted the todo.

A black screen with white text

AI-generated content may be incorrect.

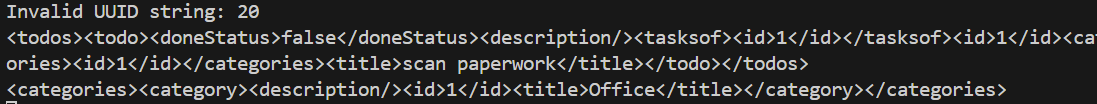
3:37 PM: Tried to delete a todo with a nonexistent ID (DELETE /todos/20). Returns a 404 error as expected.

3:40 PM: Fetching Categories for Todo with ID 1. Returns http code 200 without any other information. ## Unsure about how request headers. <http://localhost:4567/docs> does not specify

A black screen with white text

AI-generated content may be incorrect.

3:42 PM: Performed Get /todos/:id/taskof on an invalid id:20 and returned the json of ID 1 ## Bug



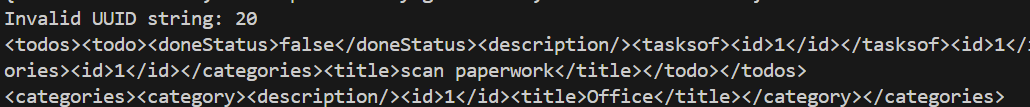
\*\* Ending session

**BUGS LOGGED**

1) Endpoint: /todos/:id/taskof, Request Type: GET

Description: Trying to get the task of an invalid id in the system. The API returned the object with Id: 1 instead.

Risk: Data concern. The user should not have access to this data.



Refer to bug summary: todoBug1.docx

**AREAS OF POTENTIAL RISK**

Data Exposure:

Bug #1: Returns data for an invalid request (nonexistent id).

Risk: Users should not have access to data for invalid requests. This could lead to unintended data exposure.

**CAPABILITIES**

CRUD operations of ToDo entity with JSON and XML.

Querying creating ToDo project relationship in XML and JSON.

Refer to Capabilities Script: ToDoXML.sh and ToDoJson.sh

**SUMMARY OF SESSION FINDINGS**

* The API supports CRUD operations for ToDo.
* The API returns status 200 for non existing relationship between TaskOf and ToDo.
* The swagger page lacks details about relationship put payload between ToDo and Category/TaskOf

**NEW TESTING IDEAS**

* Test the relationship between ToDo and projects/categories.
* Test complicated filtering requests
* Test status code when modifying nonexistent identify and verify the status of other entities to make sure that they have not been modified.