**Automation Project – Part A: Exploratory testing session #4**

**Sessions focus:** **INTEROPERABILITY**

**Charter:**

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

Identify documented and undocumented 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

**Software used for testing**

macOS Big Sur

Postman

**Testers**

Glen Xu 260767363 [jiayuan.xu@mail.mcgill.ca](mailto:jiayuan.xu@mail.mcgill.ca)

Christopher Cui 260867703 christopher.cui@mail.mcgill.ca

Duration: 39 minutes

**TEST NOTES**

Start

4:07 run the app in terminal - Running on 4567

4:08 [reading document at http://localhost:4567/docs](http://localhost:4567/docs)

4:09 click "Entities Explorer -> todo -> id:1", see two more tables

4:10 click "tasksof id:1", redirected to project

4:11 see "tasks" table, click "tasks -> id:1", redirected back to todos

4:12 click "categories id:1", redirected to category

4:12 turn on Postman, prepare for testing

4:13 [GET http://localhost:4567/todos/1/tasksof](http://localhost:4567/todos/1/tasksof)

{  
 "projects": [  
 {  
 "id": "1",  
 "title": "Office Work",  
 "completed": "false",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 },  
 {  
 "id": "2"  
 }  
 ]  
 }  
 ]  
}

4:14 [GET http://localhost:4567/todos/1/categories](http://localhost:4567/todos/1/categories)

{  
 "categories": [  
 {  
 "id": "1",  
 "title": "Office",  
 "description": ""  
 }  
 ]  
}

4:14 [GET http://localhost:4567/todos/2/tasksof](http://localhost:4567/todos/2/tasksof)

{  
 "projects": [  
 {  
 "id": "1",  
 "title": "Office Work",  
 "completed": "false",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 },  
 {  
 "id": "2"  
 }  
 ]  
 }  
 ]  
}

4:15 [GET http://localhost:4567/todos/2/categories](http://localhost:4567/todos/2/categories)

{  
 "categories": []  
}

4:15 [HEAD http://localhost:4567/todos/1/tasksof](http://localhost:4567/todos/1/tasksof)

4:16 [POST http://localhost:4567/todos/1/tasksof with the following body:  
{  
 "projects": [  
 {  
 "id": "2",  
 "title": "Home",  
 "completed": "true",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
 }  
 ]  
}](http://localhost:4567/todos/1/tasksof)

{  
 "errorMessages": [  
 "Could not find field: projects"  
 ]  
}

4:16 [tried POST http://localhost:4567/todos/1/tasksof with the following body several times:  
{  
 "id": "2",  
 "title": "Home",  
 "completed": "true",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
}](http://localhost:4567/todos/1/tasksof)

4:21 [GET http://localhost:4567/todos/1/tasksof](http://localhost:4567/todos/1/tasksof)

{  
 "projects": [  
 {  
 "id": "1",  
 "title": "Office Work",  
 "completed": "false",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 },  
 {  
 "id": "2"  
 }  
 ]  
 },  
 {  
 "id": "2",  
 "title": "",  
 "completed": "false",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
 }  
 ]  
}  
Got the new project, but the information was not updated successfully.

4:23 [DELETE http://localhost:4567/todos/1/tasksof/2](file:///Users/glenxu/Downloads/DELETE%20http:/localhost:4567/todos/1/tasksof/2)

4:25 [GET http://localhost:4567/todos/1/tasksof](http://localhost:4567/todos/1/tasksof)

{  
 "projects": [  
 {  
 "id": "1",  
 "title": "Office Work",  
 "completed": "false",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 },  
 {  
 "id": "2"  
 }  
 ]  
 }  
 ]  
}  
Deleted successfully.

4:25 [HEAD http://localhost:4567/todos/1/tasksof](http://localhost:4567/todos/1/tasksof)

4:26 [POST http://localhost:4567/todos/1/tasksof with the following body:  
{  
 "title": "TRYTHIS",  
 "completed": true,  
 "active": false,  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
}](http://localhost:4567/todos/1/tasksof)

4:27 [GET http://localhost:4567/todos/1/tasksof](file:///Users/glenxu/Downloads/GET%20http:/localhost:4567/todos/1/tasksof)

{  
 "projects": [  
 {  
 "id": "7",  
 "title": "YES",  
 "completed": "true",  
 "active": "false",  
 "description": "Here!",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
 },  
 {  
 "id": "1",  
 "title": "Office Work",  
 "completed": "false",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 },  
 {  
 "id": "2"  
 }  
 ]  
 },  
 {  
 "id": "2",  
 "title": "",  
 "completed": "false",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
 },  
 {  
 "id": "8",  
 "title": "TRYTHIS",  
 "completed": "true",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
 }  
 ]  
}  
Without id, it's added successfully.

4:28 [HEAD http://localhost:4567/projects](http://localhost:4567/projects)

4:29 [POST http://localhost:4567/projects with the following body:  
{  
 "id": "3",  
 "title": "Home",  
 "completed": true,  
 "active": false,  
 "description": "Hii!",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
}](http://localhost:4567/projects)

{  
 "errorMessages": [  
 "Invalid Creation: Failed Validation: Not allowed to create with id"  
 ]  
}

4:30 [POST http://localhost:4567/projects with the following body:  
{  
 "title": "Home",  
 "completed": true,  
 "active": false,  
 "description": "Hii!",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
}](http://localhost:4567/projects)

{  
 "id": "5",  
 "title": "Home",  
 "completed": "true",  
 "active": "false",  
 "description": "Hii!",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
}  
Wondered why the id = 5, instead of 3.

4:30 [GET http://localhost:4567/todos/1/tasksof](http://localhost:4567/todos/1/tasksof)

{  
 "projects": [  
 {  
 "id": "1",  
 "title": "Office Work",  
 "completed": "false",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 },  
 {  
 "id": "2"  
 }  
 ]  
 },  
 {  
 "id": "2",  
 "title": "",  
 "completed": "false",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
 },  
 {  
 "id": "5",  
 "title": "Home",  
 "completed": "true",  
 "active": "false",  
 "description": "Hii!",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
 }  
 ]  
}

4:31 [HEAD http://localhost:4567/projects](http://localhost:4567/projects)

4:31 [POST http://localhost:4567/projects with the following body:  
{  
 "title": "Home",  
 "completed": "true",  
 "active": "false",  
 "description": "Here!",  
 "tasks": [  
 {  
 "id": "2"  
 }  
 ]  
}](http://localhost:4567/projects)

{  
 "errorMessages": [  
 "Failed Validation: completed should be BOOLEAN, active should be BOOLEAN"  
 ]  
}

4:33 POST http://localhost:4567/projects with the following body:  
{  
 "title": "Home",  
 "completed": true,  
 "active": false,  
 "description": "Here!",  
 "tasks": [  
 {  
 "id": "2"  
 }  
 ]  
}

{  
 "id": "7",  
 "title": "Home",  
 "completed": "true",  
 "active": "false",  
 "description": "Hii!",  
 "tasks": [  
 {  
 "id": "2"  
 }  
 ]  
}  
Noticed that the id kept growing even if the last addition was not successful, it jumped over id = 6.  
It explained why we did not have id = 3 or 4.

4:33 GET http://localhost:4567/todos/1/tasksof

{  
 "projects": [  
 {  
 "id": "1",  
 "title": "Office Work",  
 "completed": "false",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 },  
 {  
 "id": "2"  
 }  
 ]  
 },  
 {  
 "id": "2",  
 "title": "",  
 "completed": "false",  
 "active": "false",  
 "description": "",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
 },  
 {  
 "id": "5",  
 "title": "Home",  
 "completed": "true",  
 "active": "false",  
 "description": "Hii!",  
 "tasks": [  
 {  
 "id": "1"  
 }  
 ]  
 }  
 ]  
}  
The added project with tasks = 2 did not get returned, which is correct.

4:34 GET http://localhost:4567/todos/1/categories

{  
 "categories": [  
 {  
 "id": "1",  
 "title": "Office",  
 "description": ""  
 }  
 ]  
}

4:35 HEAD http://localhost:4567/todos/1/categories

4:35 POST http://localhost:4567/todos/1/categories with the following body:  
{  
 "id": "2",  
 "title": "YESS",  
 "description": ""  
}

4:38 GET http://localhost:4567/todos/1/categories

{  
 "categories": [  
 {  
 "id": "1",  
 "title": "Office",  
 "description": ""  
 },  
 {  
 "id": "2",  
 "title": "Home",  
 "description": ""  
 }  
 ]  
}  
The id = 2 category was updated, but only with the original information.

4:39 DELETE http://localhost:4567/todos/1/categories/2

4:39 GET http://localhost:4567/todos/1/categories

{  
 "categories": [  
 {  
 "id": "1",  
 "title": "Office",  
 "description": ""  
 }  
 ]  
}  
The id = 2 got deleted.

4:40 GET http://localhost:4567/projects/1/tasks

{  
 "todos": [  
 {  
 "id": "1",  
 "title": "scan paperwork",  
 "doneStatus": "false",  
 "description": "",  
 "tasksof": [  
 {  
 "id": "5"  
 },  
 {  
 "id": "1"  
 },  
 {  
 "id": "2"  
 }  
 ],  
 "categories": [  
 {  
 "id": "1"  
 }  
 ]  
 },  
 {  
 "id": "2",  
 "title": "file paperwork",  
 "doneStatus": "false",  
 "description": "",  
 "tasksof": [  
 {  
 "id": "1"  
 },  
 {  
 "id": "7"  
 }  
 ]  
 }  
 ]  
}

4:42 HEAD http://localhost:4567/projects/1/tasks

4:44 POST http://localhost:4567/projects/1/tasks with the following body:  
{  
 "id": "3",  
 "title": "new todo",  
 "doneStatus": false,  
 "description": "",  
 "tasksof": [  
 {  
 "id": "1"  
 },  
 ]  
}

{  
 "errorMessages": [  
 "Could not find thing matching value for id"  
 ]  
}

4:45 POST http://localhost:4567/projects/1/tasks with the following body:  
{  
 "title": "new todo",  
 "doneStatus": false,  
 "description": "",  
 "tasksof": [  
 {  
 "id": "1"  
 },  
 ]  
}

4:46 GET http://localhost:4567/projects/1/tasks

{  
 "todos": [  
 {  
 "id": "1",  
 "title": "scan paperwork",  
 "doneStatus": "false",  
 "description": "",  
 "tasksof": [  
 {  
 "id": "5"  
 },  
 {  
 "id": "1"  
 },  
 {  
 "id": "2"  
 }  
 ],  
 "categories": [  
 {  
 "id": "1"  
 }  
 ]  
 },  
 {  
 "id": "3",  
 "title": "new todo",  
 "doneStatus": "false",  
 "description": "",  
 "tasksof": [  
 {  
 "id": "1"  
 }  
 ]  
 },  
 {  
 "id": "2",  
 "title": "file paperwork",  
 "doneStatus": "false",  
 "description": "",  
 "tasksof": [  
 {  
 "id": "1"  
 },  
 {  
 "id": "7"  
 }  
 ]  
 }  
 ]  
}

4:48 Turn off terminal and Postman.

**Bugs Logged/Closed**

- After using the POST command, the id of the capability increments no matter if the POST was successful. The capability might not have been POSTed at all, but the id still increments. If a capability was POSTed and then DELETEd, the id is still taken. This bug is inconvenient and unexpected. ##BUG (Refer to Session\_4\_Bug\_Summary.docx)

**List of Concerns Identified in Session**

- Cannot update a capability connected to another capability directly using POST, e.g. POST http://localhost:4567/projects/1/tasks/1.

- When the server is activated, todo id=1 has a relationship with category id=1, but category id=1 does not have relationships with any todo.

**List of Testing Ideas**

- try getting or posting a capability of another capability of another, aka a third level interoperability;

- further explore the POST command with controlled variables. In a capability, try POSTing another capability with id.

**Areas of Potential Risk**

- Refer to the Bugs Logged/Closed section. This bug brings confusion and inconvenience to the users. The ids that have been deleted or have not even been successfully created are still possessed by the system, and cannot be used by newly created capabilities.

**Capabilities**

~/[capability1]/:id/[capability2] GET method returns all the capability 2 items related to capability 1, with given id

~/[capability1]/:id/[capability2] HEAD method headers for the capability 2 items related to capability 1, with given id

~/[capability1]/:id/[capability2] POST method creates an instance of a relationship named capability 2 between capability 1 instance :id and the capability 2 instance represented by the id in the body of the message

~/[capability1]/:id/[capability2] **/:id** Delete method deletes the instance of the relationship named between capability 1 and capability 2 using the :id