**Automation Project – Part A: Exploratory testing session #3**

**Sessions focus: Categories**

**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**

Windows 10

Postman

**Testers**

Bowen Li 260787692 [bo.w.li@mail.mcgill.ca](mailto:bo.w.li@mail.mcgill.ca)

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

Duration: 45 minutes

**TEST NOTES**

Start

4:49 Launch application on command line

4:50 Application is running on 4567, http://localhost:4567

4:51 Open <http://localhost:4567/docs> on browser

4:52 Documentation open, start reading documentation

4:55 Open Postman and get ready to start testing the /categories endpoints

4:57 GET request using <http://localhost:4567/categories>

JSON return body:

{

"categories": [

{

"id": "1",

"title": "Office",

"description": ""

},

{

"id": "2",

"title": "Home",

"description": ""

}

]

}

Expected capability. Found 2 default categories.

5:00 HEAD request using <http://localhost:4567/categories>

JSON return content type:

application/json

Expected capability.

5:02 POST request using <http://localhost:4567/categories>

JSON input body:

{

"id": "3",

"title": "School",

"description": "school related description"

}

JSON return body:

{

"errorMessages": [

"Invalid Creation: Failed Validation: Not allowed to create with id"

]

}

Expected failure. Documentation does not include id.

5:04 POST request with <http://localhost:4567/categories>

JSON input body:

{

"title": "School",

"description": "school related description"

}

JSON return body:

{

"id": "3",

"title": "School",

"description": "school related description"

}

Expected capability. Assigned a new category with id 3.

5:06 POST request with <http://localhost:4567/categories>

JSON input body:

{

"title": "School2",

"description": "school2 related description"

}

JSON return body:

{

"id": "4",

"title": "School2",

"description": "school2 related description"

}

Expected capability. Assigned new category with id 4.

5:09 GET request with <http://localhost:4567/categories>

JSON return body:

{

"categories": [

{

"id": "4",

"title": "School2",

"description": "school2 related description"

},

{

"id": "3",

"title": "School",

"description": "school related description"

},

{

"id": "1",

"title": "Office",

"description": ""

},

{

"id": "2",

"title": "Home",

"description": ""

}

]

}

Expected capability. 2 default categories and 1 new recently added category returned.

5:12 GET request with <http://localhost:4567/categories/3>

JSON return body:

{

"categories": [

{

"id": "3",

"title": "School",

"description": "school related description"

}

]

}

Expected capability. Appropriate category returned.

5:15 HEAD request with <http://localhost:4567/categories/3>

JSON return content type:

application/json

Expected capability.

5:17 POST request with <http://localhost:4567/categories/3>

JSON input body:

{

"title": "School2",

"description": "school2 related description"

}

JSON return body:

{

"title": "School2",

"description": "school2 related description"

}

Expected capability. Title and description changed.

5:19 GET request with <http://localhost:4567/categories/3>

JSON return body:

{

"categories": [

{

"id": "3",

"title": "School2",

"description": "school2 related description"

}

]

}

Expected capability. Confirms the title and description change from previous command.

5:21 POST command with <http://localhost:4567/categories/5>

JSON input body:

{

"title": "School",

"description": "school related description"

}

JSON output body:

{

    "errorMessages": [

        "No such category entity instance with GUID or ID 5 found"

    ]

}

Unexpected failure. Cannot create new categories with specified id in endpoint.

5:24 PUT command with <http://localhost:4567/categories/3>

JSON input body:

{

"title": "School3",

"description": "school3 related description"

}

JSON output body:

{

"id": "3",

"title": "School3",

"description": "school3 related description"

}

Expected capability. Title and description changed.

5:27 DELETE command with <http://localhost:4567/categories/4>

Status code: 200 OK

Expected capability. Can delete categories with valid id.

5:29 GET command with <http://localhost:4567/categories/4>

Status code: 404 NOT FOUND

Expected capability. Confirms category is deleted

5:32 POST command with <http://localhost:4567/categories>

JSON input body:

{

"title": "School",

"description": "school related description"

}

JSON output body:

{

"id": "5",

"title": "School",

"description": "school related description2"

}

Expected capability. Discovered that after deleting a category at an id, then adding a new category, the unused id is not used, but is incremented.

5:38 Close command line and Postman

**Bugs Logged/Closed**

- After performing POST /categories then DELETE /categories/id to delete the new category and then POST /categories again, the id of the new category is not the id as the deleted category, but one incremented from it. This is unexpected and inconvenient behavior. ##BUG (Refer to Session\_3\_Bug\_Summary.docx)

**List of Concerns Identified in Session**

- Cannot specify id of a category when creating a category with /categories endpoint POST method

- Cannot create new categories at new ids with /categories/:id endpoint POST method

**List of Testing Ideas**

- explore testing /categories/:id endpoint PUT method with only title or only description

- explore testing /categories/:id endpoint POST method with only title or only description

**Areas of Potential Risk**

- Refer to the Bugs Logged/Closed section, the bug presents inconvenience to users as the deleted id can never be used again and may cause confusion to users as they did not expect the new category to have an id of the deleted id incremented by one.

**Capabilities**

- /categories endpoint GET method returns all existing categories

- /categories endpoint HEAD method returns appropriate header information

- /categories endpoint POST method creates a new category and assigns a new id to it

- /categories/:id endpoint GET method returns category at specified id

- /categories/:id endpoint HEAD method returns appropriate header information for specified category at id

- /categories/:id endpoint PUT method edits the title and description of a specified category by id

- /categories/:id endpoint POST method edits the title and description of a specified category by id

- /categories/:id endpoint DELETE method deletes an existing category at a specified id