Session Notes 3 – ECSE 429 (Categories): Vladimir

**Charter**

* Identify capabilities and areas of potential instability of the “REST API todo list manager.”
* Identify documented and undocumented capabilities for Todo & Project Categories.
* Create and test scripts for each capability.
* Exercise each capability with data typical to intended use.

**Build**

* runTodoManagerRestAPI-1.5.5.jar

**Area**

* Categories:

**Test Environment**

* Windows 10
* Postman v11
* API running locally

**Date & Time**

* October 2nd, 2025 – Start time: 2:30pm
* Session duration: 45 minutes

**Tester**

* Vladimir Venkov – 261040180 – [vladimir.venkov@mail.mcgill.ca](mailto:vladimir.venkov@mail.mcgill.ca)

**Session Log**

**2 :30pm**

GET /categories. Response: 200 OK with list of categories:

[{id: 1, title: Office}, {id: 2, title: Home}]

Note: Categories returned are not sorted by id.

A screenshot of a computer

AI-generated content may be incorrect.

**2:32pm**

HEAD /categories. Response 200 OK with empty response message

**A screenshot of a computer

AI-generated content may be incorrect.**

**2:33pm**

POST /categories. Create a new category with {“title”: “School”, “description”: “”}. Response 201 Created.

Note: id (3) was auto-generated and auto-incremented.

A screenshot of a computer

AI-generated content may be incorrect.

**2:35pm**

GET /categories/3. Response 200 OK with list of categories containing one entry: previously created School category.

A screenshot of a computer

AI-generated content may be incorrect.

Note: in the case where and id is not found, error 404 is returned, however, the error message is not entirely clear.  
A screenshot of a computer

AI-generated content may be incorrect.

**2:36pm**

HEAD /categories/3. Response 200 OK with empty message body.

A screenshot of a computer

AI-generated content may be incorrect.

**2:37pm**

POST /categories/3.

Note: providing id in request body causes error 400, however, error message is unclear:

A screenshot of a computer program

AI-generated content may be incorrect.

Error message should be something like: “id of category cannot be amended.”

When id is not provided, endpoint works as expected:

A screenshot of a computer

AI-generated content may be incorrect.

**2:41pm**

PUT /categories/3

Note: Similar behavior to POST /categories/3, API return 400 if id is provided in request body.

A screenshot of a computer program

AI-generated content may be incorrect.

Correct behavior when id is not provided in request body.

A screenshot of a computer program

AI-generated content may be incorrect.

Note: The behavior for this endpoint is the exact same as POST, leading to confusing API design.

**2:45pm**

DELETE /categories/3 allows to delete a category.

A screenshot of a computer

AI-generated content may be incorrect.

Once again, if the id does not exist, an error 404 is thrown:

A screenshot of a computer

AI-generated content may be incorrect.

**2:49pm**

GET /categories/1/todos returns 200 OK

A screenshot of a computer

AI-generated content may be incorrect.

However, this is a wrong response:  
A screenshot of a computer

AI-generated content may be incorrect.

When fetching all todos, we see that todo with id 1 does have a relationship with category with id 1.

Another problem is when fetching the todos of a category that does not exist:

A screenshot of a computer

AI-generated content may be incorrect.

This returns an empty list instead of throwing error 404, which would be the expected response.

**2:51pm**

HEAD /categories/2/todos returns 200 OK with empty response body

A screenshot of a computer

AI-generated content may be incorrect.

**2:53pm**

POST /categories/1/todos with request body: {“id”: 1} return error 404

A screenshot of a computer program

AI-generated content may be incorrect.

According to API documentation, this should create a relationship between category 1 and todo 1, which both exist.

Note: If instead of providing id, we send a request body like so: {“title”: “todo\_1”}, it return 201 Created and a new todo is created.

A screenshot of a computer

AI-generated content may be incorrect.

However, this todo does not have a relationship with the category:

A screenshot of a computer

AI-generated content may be incorrect.

But it does show up in the list of todos when using GET /categories/1/todos:

A screenshot of a computer

AI-generated content may be incorrect.

**3:03pm**

DELETE /categories/1/todos/3 returns 200 OK.

A screenshot of a computer

AI-generated content may be incorrect.

And the todo is indeed removed from the list of todos of category 1:

A screenshot of a computer

AI-generated content may be incorrect.

**3:04pm**

GET /categories/1/projects returns a response of 200 OK.

A screenshot of a computer

AI-generated content may be incorrect.

**3:06pm**

HEAD /categories/1/projects returns a response of 200 OK

A screenshot of a computer

AI-generated content may be incorrect.

**3:07pm**

POST /categories/1/projects with a request body {“id”: 1} once again returns a response of error 404, even though the API documentation states that this is the intended use case for this endpoint.

A screenshot of a computer

AI-generated content may be incorrect.

And once again, if we use a request body including a title: {“title”: “my\_project”}, a new project is created with a relationship to category 1.

A screenshot of a computer program

AI-generated content may be incorrect.

**3:12pm**

DELETE /categories/1/projects/1 properly deletes the relationship and returns 200 OK.

A screenshot of a computer

AI-generated content may be incorrect.

**Session Summary Findings**

**Capabilities**

* CRUD operations work reliably on Categories objects.
* Target ID is validated for existence on Update and Delete operations.
* IDs are auto-incremented and auto-generated.
* HEAD operations work on objects.

**Key Issues**

* Error Messages
  + Unclear error messages
  + No difference between bad usage and missing object/target
  + Messages are not human friendly making black-box debugging of the API harder
* Relationship Handling is not operational / inconsistent when using categories endpoints
  + Relationships cannot be created between existing categories and todos or projects through the /categories endpoint.
  + Only relationships between existing categories and new todos or projects can be created through the /categories endpoint.
* Confusing API design
  + The /categories/id endpoint allows for a POST and PUT method, which act in the exact same way.
  + This creates confusion for the user.
  + The POST method should be disallowed.

**New Test Ideas**

* Test for extremely large (9999999999999) or negative ids (-1) in endpoints.
* Test for unexpected fields in request body.
* Try deletion of category with existing todos or projects that were created through the categories endpoint.

**Areas of Potential Risk**

* Problems and inconsistencies in relationship handling: Established relationships could sometimes be accessible only through todos or categories.
* Error message clarity: Unclear error messages could leave users confused.
* Lack of data validation: Categories with empty descriptions or duplicate titles could lead to confusion

**Files Referenced**

* Postman Collection used for categories tests.