Bug Summary

BUG I

<u>Execution Summary</u>: Inconsistency with `POST /(todos;projects;categories)/:id` Endpoint.
Failed test signatures.

<u>Bug Summary</u>: The *`POST /(todos;projects;categories)/:id`* endpoint behaves similarly to the *`PUT`* method instead of adhering to its intended CRUD definition of "Creating."

<u>Impact</u>: Without having the API documentation, it is easier to make new users confused and not able to use POST function properly.

<u>Reproduction Steps</u>:

- Executed *`POST /todos/:id` or `POST /projects/:id` or `POST /categories/:id`* with payload *`{title: "new title"}`*, and *':id'* is the id of an existing object.
  - Expected Result: Creation of a new object.
  - Actual Result: Updated the title of the existing object and returned a status code of 200.
- Executed *`POST /todos/:id` or `POST /projects/:id`* or *`POST /categories/:id`* with payload *`{title: "new title"}`*, and *':id'* is the id of a not existing object.
  - Expected Result: Creation of a new object.
  - Actual Result: Error message with a status code of 404.

BUG II

<u>Execution Summary</u>: Inaccurate Responses for *`GET / (todos;projects;categories) /:id/childNodes`* Endpoints.

<u>Bug Summary</u>: The endpoints do not return accurate feedback for non-existent parent node (*todos;projects;categories)* entities.

<u>Impact</u>: Using GET function to get child nodes, which under one of three parent nodes, may reflect the wrong information of the system, that provides the user has an illusion of the system is full of todos/projects/categories projects.

<u>Reproduction Steps</u>:

- Execute *`GET / (todos;projects;categories) /:id/childNodes`*, and ':id' is the id of a non-existing object in three main objects.
  - Expected Result: Error message with a status code of 404.
  - Actual Result: No response with a status code of 200.

BUG III

<u>Execution Summary</u>: Anomaly in *`POST / (todos;projects;categories) /:id/childNodes`* Endpoint.

<u>Bug Summary</u>: the endpoint seems to exhibit unexpected behavior by skipping the creation of certain category objects.

<u>Impact</u>: It may let the user wrongly believe that there are more child nodes that are prior to the newly created child nodes. (The user will realize it when running *`GET / (todos;projects;categories) /:id/childNodes`*

<u>Reproduction Steps</u>:

- Execute *`POST / (todos;projects;categories) /:id/childNodes`* with payload *`{title: "new child"}`*, and then execute *`GET / (todos;projects;categories) /:id/childNodes`*.
  - Expected Result: after 'GET', the 'id' of the new child node (with name "new child") is equal to the number of child nodes shown on screen, which means the creation of a new child node is sequential.

    - Actual Result: the 'id' of new child node (with name "new child") is LARGER than the number of child nodes shown on screen. Therefore, the creation of a new child node is not sequential, and some child nodes' ids have been escaped.

Specific example:

- Executed `*POST /todos/1/categories*` with payload `*{title: "new category"}*`.
    - Expected Result: Creation of a new category object sequentially.
    - Actual Result: Created a new category object with id=3 and title = "new category" with a status code 201.
- Executed `*GET /todos/1/categories*`.
    - Result: Retrieved two categories with id=1 and id=3, implying that the creation process for `*POST /todos/:id/categories*` missed creating the category object with id=2.

BUG IV

Execution Summary: inconsistency behavior between method not allowed

Bug Summary: For some undocumented tests we did, the system provides feedback of "404 not found". Because the endpoints are undocumented, the correct feedback should be "405 method is not allowed".

Impact: It may provide wrong feedback to user, that lead user to find the problem of their inputs instead of self-examination about whether this input is allowed.

Reproduction Steps:

Use specific HTTP Method to test endpoint. And test the retuned status code. Issues founded in the following testcases, expected status code is 405, while the returned state ode is 404:

```
test_todos_id_catrgories_id_notallow_get
        test_todos_id_catrgories_id_notallow_post
        test_todos_id_catrgories_id_notallow_head
        test_todos_id_tasksof_id_notallow_get
        test_todos_id_tasksof_id_notallow_post
        test_todos_id_tasksof_id_notallow_head
```

BUG V

Execution Summary: API not compatible with xml

Bug Summary: API input should work and return the code of success (200 for GET, 201 for POST), but it fails to process XML body and returns not found status code (404).

Impact: It banned users from using XML body to input and limited the user only to use JSON body for input.

Reproduction Steps:

Request the endpoint with XML header and XML body, the response is not as expected, or similar to JSON body, this happened in the following test cases:

```
        test_post_project_xml_xml
        test_project_task_post_xml_xml
        test_project_categories_xml_json
        test_todos_xml_post_xml_json
        test_todos_xml_post_xml_xml
```