**ECSE 429:**
**October 11th, 2021**
**David Kronish 260870097**
**Abrar Fahad Rahman Anik 260851928**

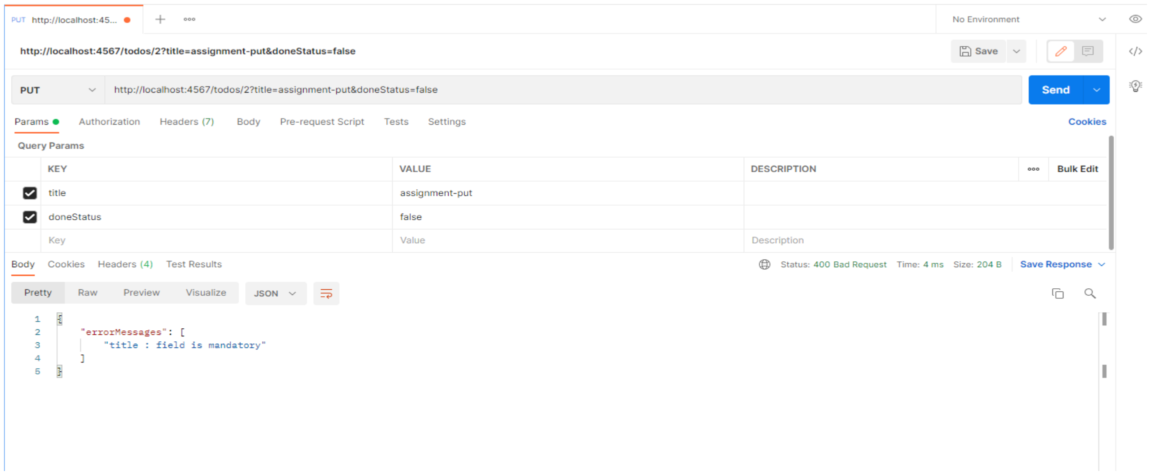
# Assignment II: Exploratory Testing Final Findings Report

In both of our exploratory testing sessions, we used a strong style pairing approach to yield increased engagement and a larger coverage for the functions tested. In session 1, we covered the capabilities of /todos, whereas in session 2 we covered the capabilities of /categories. Throughout our testing process, we discovered many bugs in the system, and more often than not these bugs we're either similar or associated with each other. This wasn't a surprise to us, as we both understand that the interoperability of a system is crucial to it's performance. If a single component is failing, this effect is most likely represented somewhere else in the system. In order to reduce the repetitiveness of this document, we have classified these similar bugs into two categories:

1. POST/PUT Methods
2. GET Methods for IDs that don't exist.

***POST/PUT methods***

In the rest API todo list manager application, for the capabilities under /todos, none of the POST/PUT methods worked. They either gave a wrong error message, or the data never got updated. This happened every time such a method was called, making it difficult to further test the capability with new instances of todos, projects, and categories. The following screenshots show the two different kinds of bugs found using the POST and PUT methods.



http://localhost:4567/todos/2?title=assignment-put&doneStatus=false

PUT http://localhost:4567/todos/2?title=assignment-put&doneStatus=false

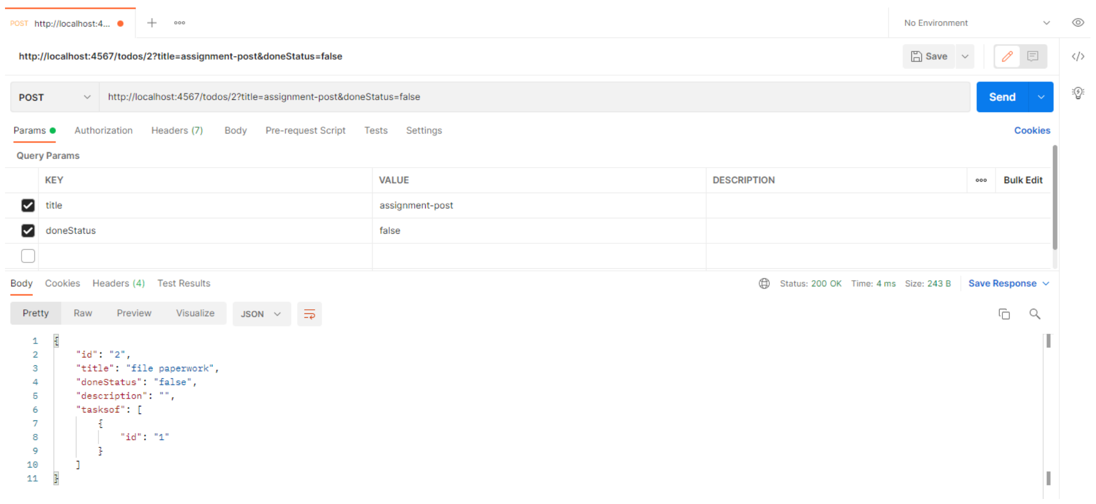
Query Params

| | KEY | VALUE | DESCRIPTION |
|---|---|---|---|
| ✓ | title | assignment-put | |
| ✓ | doneStatus | false | |
| | Key | Value | Description |

Status: 400 Bad Request Time: 4 ms Size: 204 B

```
{
    "errorMessages": [
        "title : field is mandatory"
    ]
}
```

*Image 1.1: Bug found from PUT method*


POST http://localhost:4...

No Environment

http://localhost:4567/todos/2?title=assignment-post&doneStatus=false

Save

POST http://localhost:4567/todos/2?title=assignment-post&doneStatus=false Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

| | KEY | VALUE | DESCRIPTION |
|---|---|---|---|
| ✓ | title | assignment-post | |
| ✓ | doneStatus | false | |
| | | | |

Body Cookies Headers (4) Test Results Status: 200 OK Time: 4 ms Size: 243 B Save Response

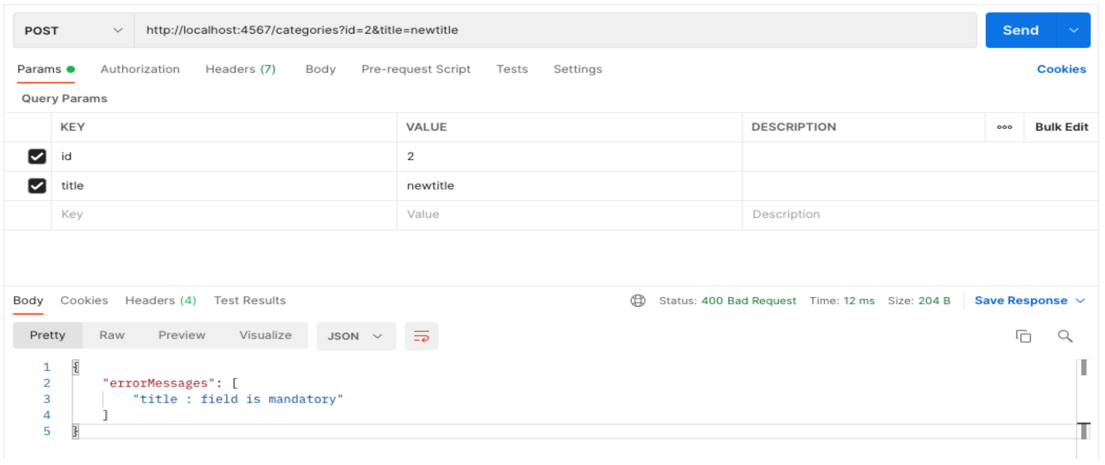
Pretty Raw Preview Visualize JSON

```
{
    "id": "2",
    "title": "file paperwork",
    "doneStatus": "false",
    "description": "",
    "tasksof": [
        {
            "id": "1"
        }
    ]
}
```

*Image 1.2: Bug found from POST method*

Even though the query parameters were passed correctly as shown in the documentation, we received false errors and or the data was not updated at all for either of the methods and returned with status 200 OK. We marked this as a major bug, as it is an important functionality of the application.

This bug was also present in POST methods for /categories (*Shown in figure 1.3*). Via it's specification, in order to create an instance of a category, only a title field is mandatory. In the example below, a POST request is being made with query parameters id and title. The expected result of this request would be an updated instance with id = 2 and a new value in the title field 'newTitle'. The bug causes the request to result in a 400 Bad Request and an error message returned. This needs to be rectified immediately as the creation and update of category instances is paramount to this application's functionality. This behaviour is also present in PUT method requests for /categories.


POST http://localhost:4567/categories?id=2&title=newtitle Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

| | KEY | VALUE | DESCRIPTION |
|---|---|---|---|
| ✓ | id | 2 | |
| ✓ | title | newtitle | |
| | Key | Value | Description |

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 12 ms Size: 204 B Save Response

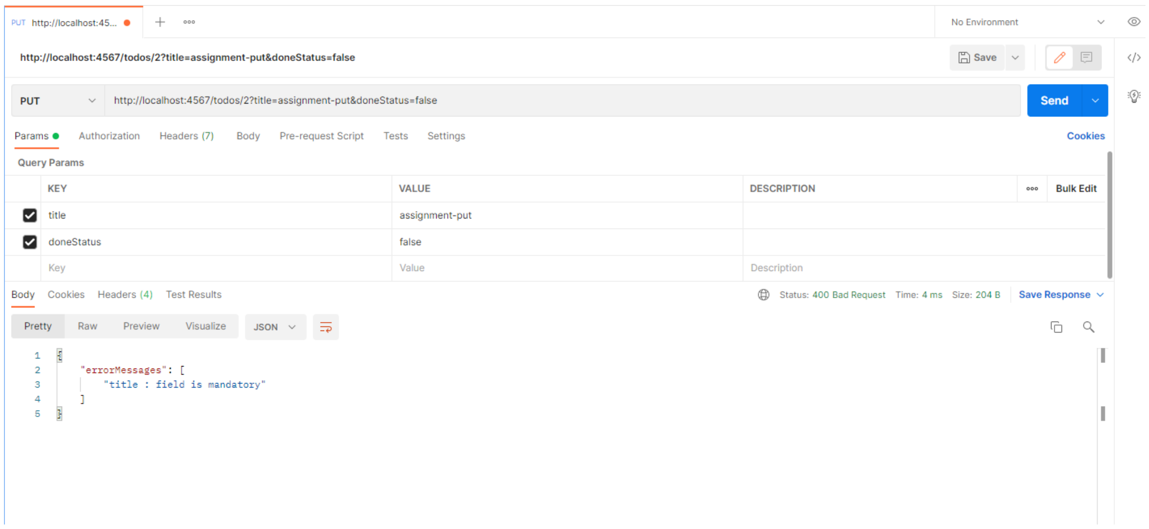
Pretty Raw Preview Visualize JSON

```
{
    "errorMessages": [
        "title : field is mandatory"
    ]
}
```

*Image 1.3: Bug found from POST method*

### *GET methods for IDs that do not exist*

The second class of bugs we found when performing GET methods with /todos and /todos/:id/categories. Here, when we used the method to get an instance of todo or category using a non-existing ID, we got a status 200 OK and an empty list of the class. We found this to be misleading, as it should give a status code of 404 not found with an error message to inform the user that the instance with :id does not exist.



PUT http://localhost:4567/todos/2?title=assignment-put&doneStatus=false

Query Params

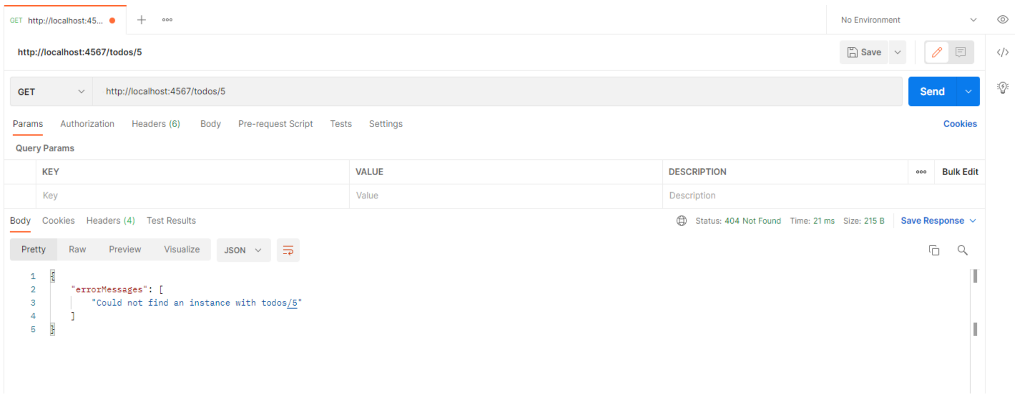
| | KEY | VALUE | DESCRIPTION |
|---|---|---|---|
| ✓ | title | assignment-put | |
| ✓ | doneStatus | false | |

Status: 400 Bad Request Time: 4 ms Size: 204 B

```
{
    "errorMessages": [
        "title : field is mandatory"
    ]
}
```

Image 1.4: Bug found from GET method

In the GET method for todos/:id, the method was implemented correctly. As you can see in the following screenshot, performing GET with [http://localhost:4567/todos/5](http://localhost:4567/todos/5) gave a sensible error message with status 404 not found. The two methods mentioned earlier should follow the error handling implementation for this method.



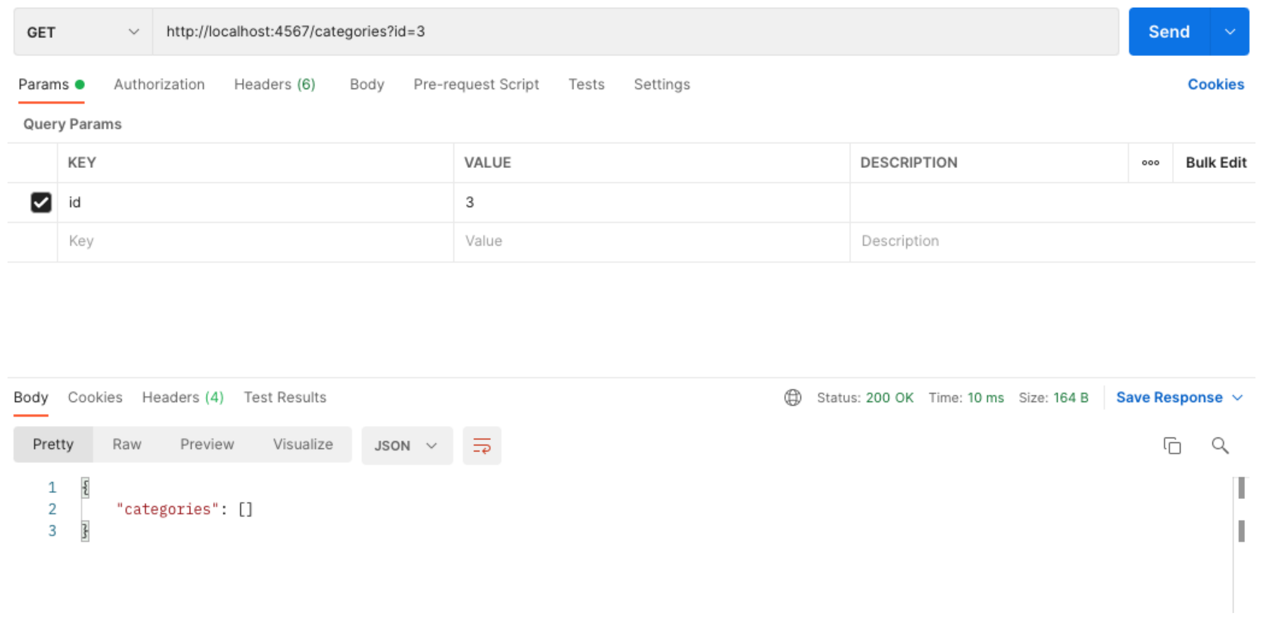
GET http://localhost:4567/todos/5

Status: 404 Not Found Time: 21 ms Size: 215 B

```
{
    "errorMessages": [
        "Could not find an instance with todos/5"
    ]
}
```

Image 1.5: Correct output from a GET method

This problem is also present in GET Methods for the /categories endpoint. As shown in image 1.6, a GET method is used along with a query parameter for ID = 3 to retrieve an instance of category. However, an instance with ID = 3 doesn't exist in the system. Regardless, the request returns an empty list of categories and a 200 OK status code. This is the wrong status code and wrong return, the request should return a 404 Not Found Error to properly indicate to the requester that there are no instances. Without this functionality, the system is severely compromised as there can be many situations where the presence or absence of an instance will be crucial to know. This should be rectified with proper status codes.



```
http://localhost:4567/categories?id=3
```

| KEY | VALUE | DESCRIPTION |
|---|---|---|
| id | 3 | |
| Key | Value | Description |

Status: 200 OK Time: 10 ms Size: 164 B

```
{
    "categories": []
}
```

*Image 1.6: BUG from a GET method*