SCHOOL OF ARTS AND SCIENCES

DEPARTMENT OF COMPUTER, INFORMATION SCIENCES AND MATHEMATICS

API Documentation

Presented to the Faculty of the

Department of Computer, Information Sciences and Mathematics

University of San Carlos

In Partial Fulfillment

of the Requirements for the Course

IT 3206 - Integrative Programming

By

Lorenz Gil Suico, Francis Louis Aliser,

Nicholai Julian Oblina, James Winston Ng

Ian Lemuel Caballes, Gerald Glenn Silverio

Allen Louie Dano, Clyde Joseph Noob

Chris Ray Belarmino

Instructor

May 15, 2024

Title of Content

API Documentation

Create

1. Create New Task

    Method: POST

    Sample API Link: http://localhost:8080/api/tasks

    JSON Data:

    ```
    {
        "title": "Sample Title",
        "description": "Sample Description"
    }
    ```

    Sample Response:

    ```
    {
        "status": 200,
        "success": true,
        "taskData": {
            "title": "Sample Title",
            "description": "Sample Description"
        },
        "message": "Task Created!"
    }
    ```
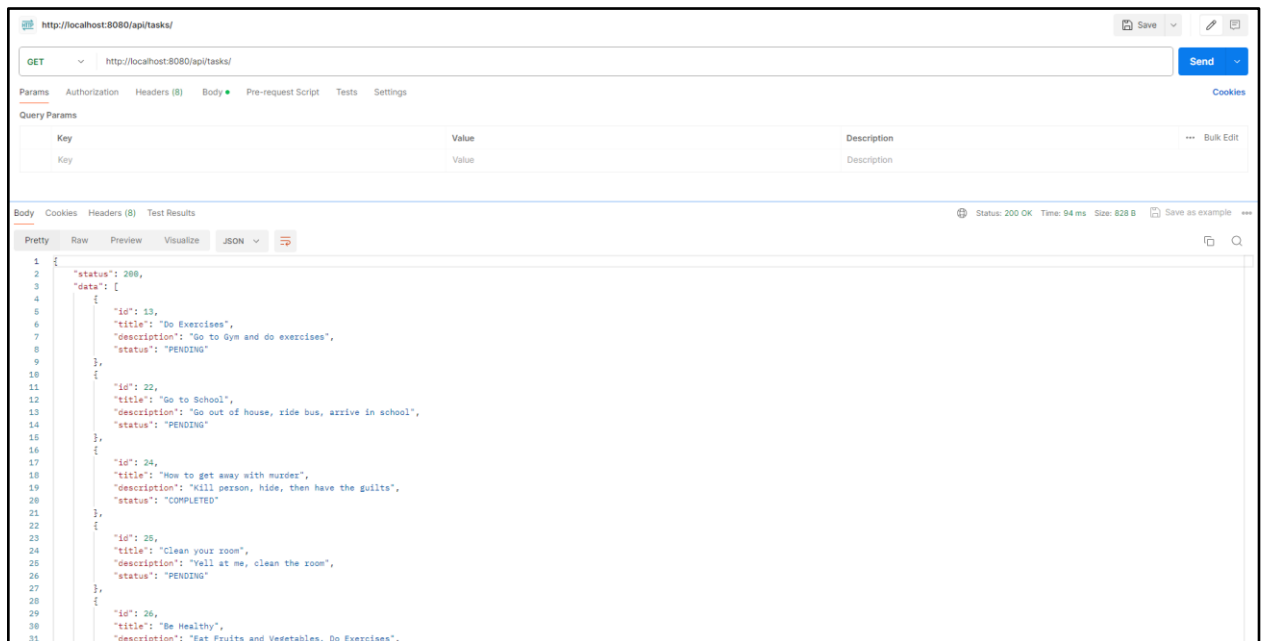
    Postman:

Read

1. Fetch All Tasks

   Method: GET

   Sample API Link: http://localhost:8080/api/tasks/

   Sample Response:

```
{
    "status": 200,
    "data": [
        {
            "id": 13,
            "title": "Do Exercises",
            "description": "Go to Gym and do exercises",
            "status": "PENDING"
        },
        {
            "id": 22,
            "title": "Go to School",
            "description": "Go out of house, ride bus, arrive in school",
            "status": "PENDING"
        },
        {
            "id": 24,
            "title": "How to get away with murder",
            "description": "Kill person, hide, then have the guilts",
            "status": "COMPLETED"
        },
        {
            "id": 25,
            "title": "Clean your room",
            "description": "Yell at me, clean the room",
            "status": "PENDING"
        },
        {
            "id": 26,
            "title": "Be Healthy",
            "description": "Eat Fruits and Vegetables, Do Exercises",
            "status": "PENDING"
        }
    ]
}
```
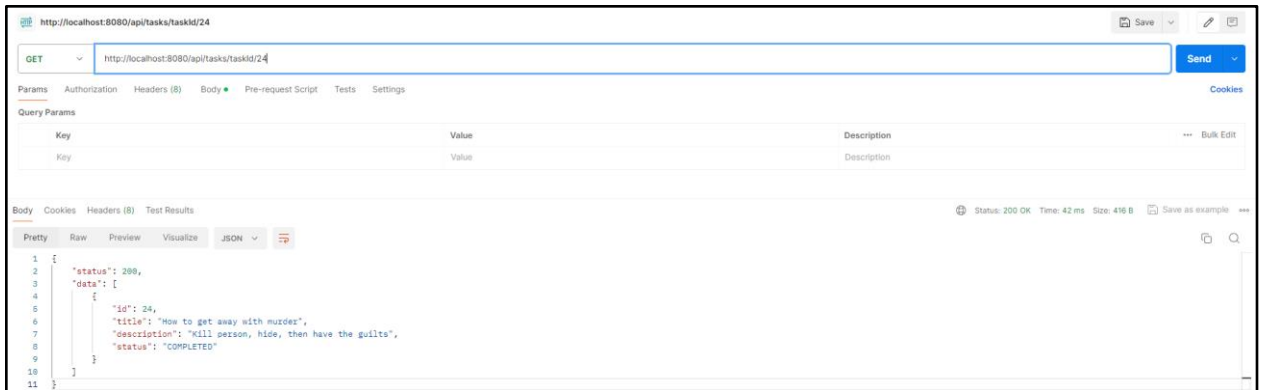
Postman

2. Fetch Tasks by ID

Method: GET

Sample API Link: http://localhost:8080/api/tasks/taskId/24

Sample Response:

```
{
    "status": 200,
    "data": [
        {
            "id": 24,
            "title": "How to get away with murder",
            "description": "Kill person, hide, then have the guilts",
            "status": "COMPLETED"
        }
    ]
}
```

Postman:

3. Fetch Task By Title

Method: GET

Sample API Link: http://localhost:8080/api/tasks/taskTitle/Do Exercises

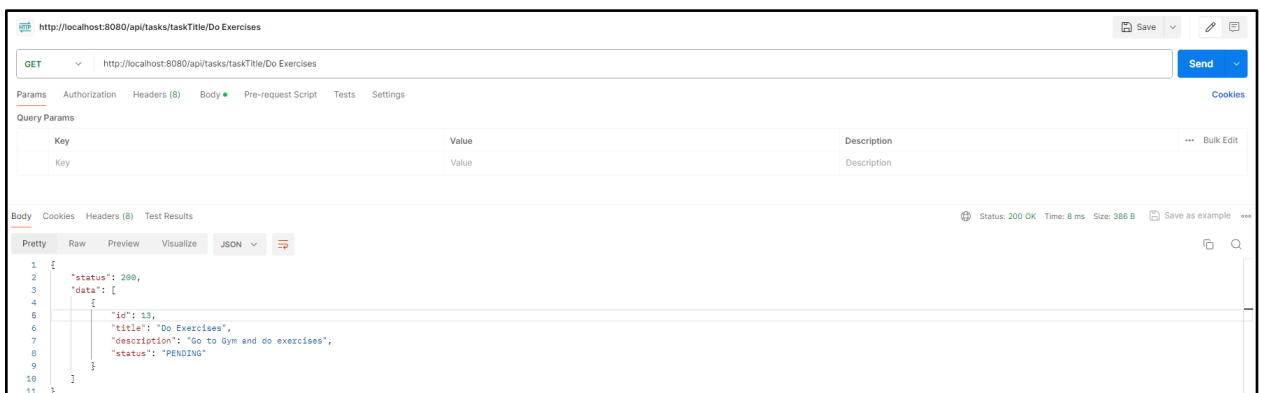Sample Response:

```
{
    "status": 200,
    "data": [
        {
            "id": 13,
            "title": "Do Exercises",
            "description": "Go to Gym and do exercises",
            "status": "PENDING"
        }
    ]
}
```
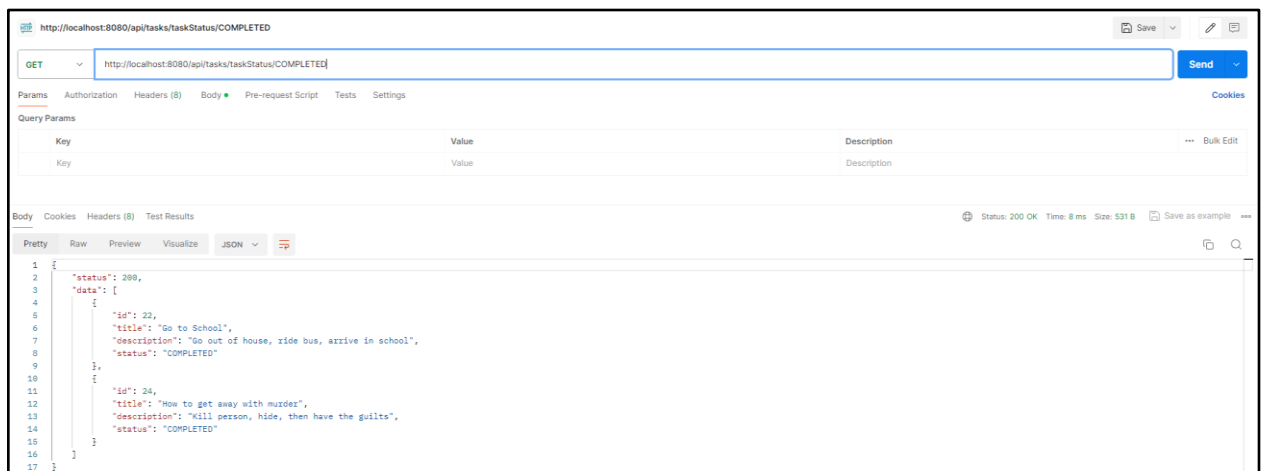
Postman:

4. Fetch Task by Status

Method: GET

Sample API Link: http://localhost:8080/api/tasks/taskStatus/COMPLETED

Sample Response:

```json
{
    "status": 200,
    "data": [
        {
            "id": 22,
            "title": "Go to School",
            "description": "Go out of house, ride bus, arrive in school",
            "status": "COMPLETED"
        },
        {
            "id": 24,
            "title": "How to get away with murder",
            "description": "Kill person, hide, then have the guilts",
            "status": "COMPLETED"
        }
    ]
}
```

Postman:

Update

1. Update a Task

    Method: PUT

    Sample API Link:http://localhost:8080/api/tasks/28
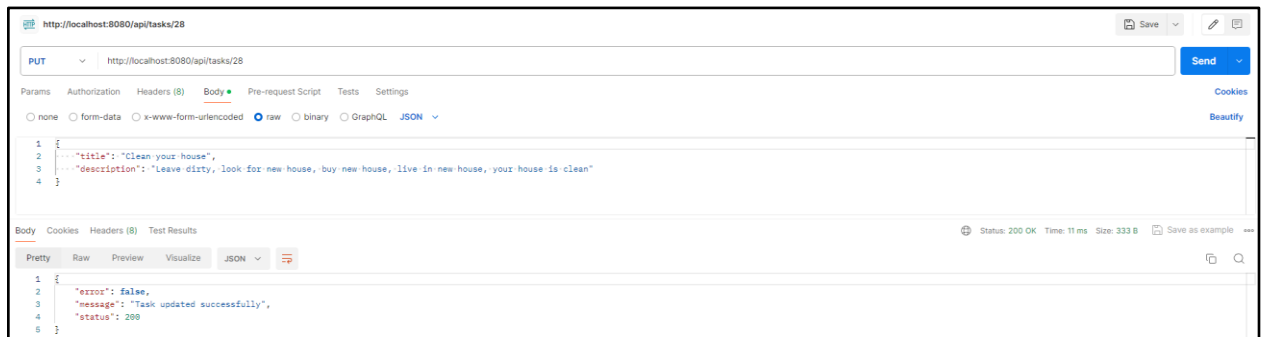
    JSON Data:

    ```
    {
        "title": "Clean your house",
        "description": "Leave dirty, look for new house, buy new house, live in new house, your house is clean"
    }
    ```

    Sample Response:

    ```
    {
        "error": false,
        "message": "Task updated successfully",
        "status": 200
    }
    ```

    Postman:

2. To start a task

    Method: PUT

    Sample API Link: http://localhost:8080/api/tasks/startTask/28

    Sample Response:

    ```
    {
        "error": false,
        "message": "Task started successfully",
        "status": 200
    }
    ```

    Postman:

3. To finish a task

Method: PUT

Sample API Link: http://localhost:8080/api/tasks/finishTask/28

Sample Response:
```
{
    "error": false,
    "message": "Task ended successfully",
    "status": 200
}
```

Postman:

Delete

1. Delete a task

   Method: DELETE

   Sample API Link: http://localhost:8080/api/tasks/28

   Sample Response:

   ```
   {
       "error": false,
       "message": "Task Deleted!",
       "status": 200
   }
   ```

   Postman:

2. Delete Multiple Task

   Method: POST

   Sample API LInk: http://localhost:8080/api/tasks/deleteMultipleTasks

   JSON Data:

   ```
   {
       "taskIds": [22,25]
   }
   ```

   Sample Response:

   ```
   {
       "error": false,
       "message": "Selected Tasks deleted successfully",
       "status": 200
   }
   ```

   Postman:

http://localhost:8080/api/tasks/deleteMultipleTasks

| Save | ✎ | 💬 |

POST | http://localhost:8080/api/tasks/deleteMultipleTasks | Send ▾

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings   Cookies

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON ▾   Beautify

```
1   [
2       "taskIds": [22,25]
3   }
```

Body Cookies Headers (8) Test Results   ⊕ Status: 200 OK Time: 24 ms Size: 343 B   ⊟ Save as example •••

Pretty Raw Preview Visualize JSON ▾ ⇶   ⧉ 🔍

```
1   {
2       "error": false,
3       "message": "Selected Tasks deleted successfully",
4       "status": 200
5   }
```