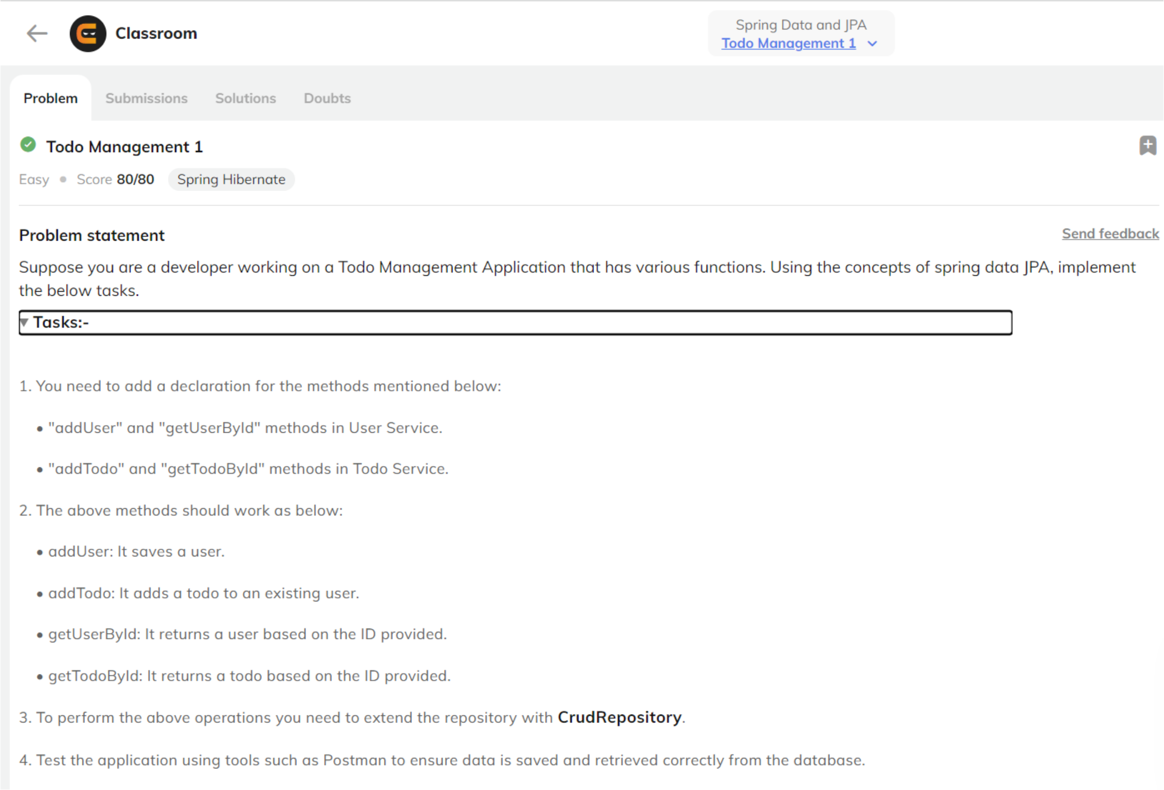
Todo Management I

Problem Statement:



Todo Management 1

Easy • Score 80/80 Spring Hibernate

Problem statement

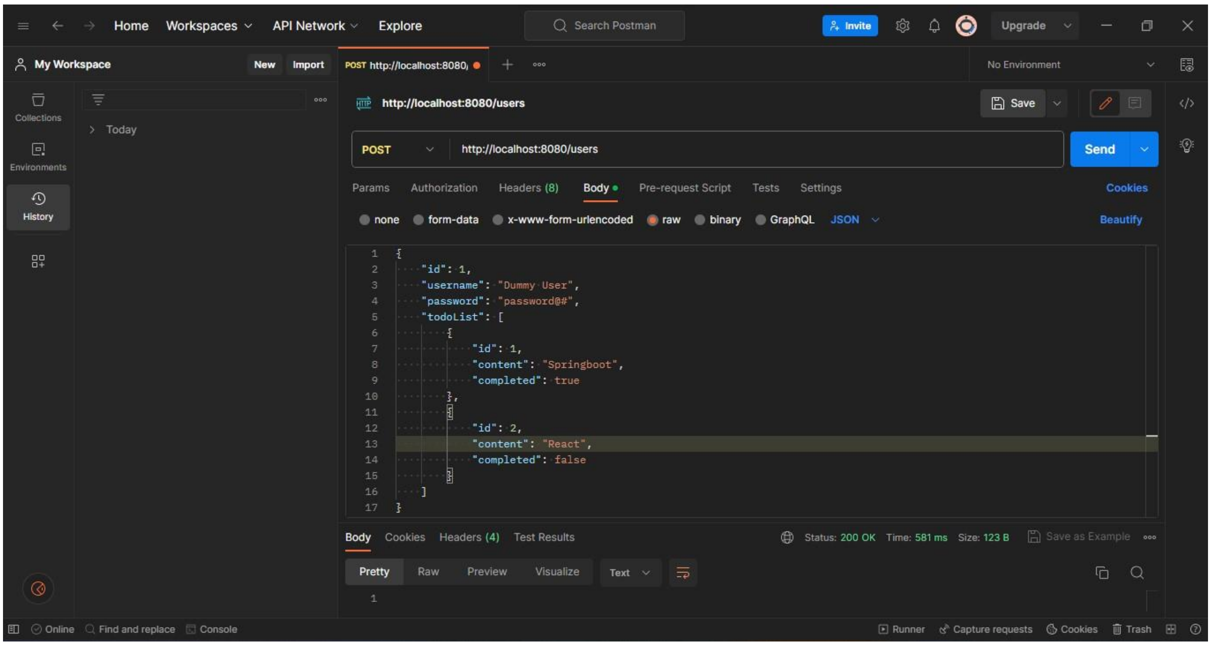
Send feedback

Suppose you are a developer working on a Todo Management Application that has various functions. Using the concepts of spring data JPA, implement the below tasks.

▾ Tasks:-

1. You need to add a declaration for the methods mentioned below:
   - "addUser" and "getUserById" methods in User Service.
   - "addTodo" and "getTodoById" methods in Todo Service.
2. The above methods should work as below:
   - addUser: It saves a user.
   - addTodo: It adds a todo to an existing user.
   - getUserById: It returns a user based on the ID provided.
   - getTodoById: It returns a todo based on the ID provided.
3. To perform the above operations you need to extend the repository with **CrudRepository**.
4. Test the application using tools such as Postman to ensure data is saved and retrieved correctly from the database.

Output:

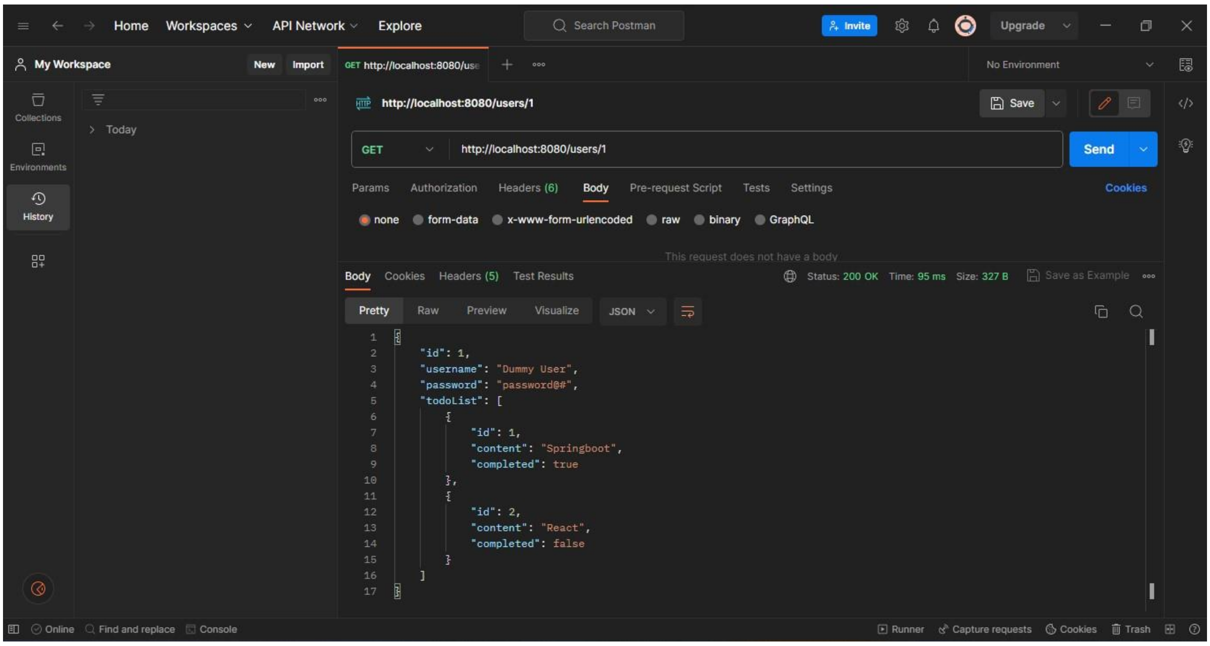


POST http://localhost:8080/users

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "id": 1,
    "username": "Dummy User",
    "password": "password@#",
    "todoList": [
        {
            "id": 1,
            "content": "Springboot",
            "completed": true
        },
        {
            "id": 2,
            "content": "React",
            "completed": false
        }
    ]
}
```

Status: 200 OK Time: 581 ms Size: 123 B



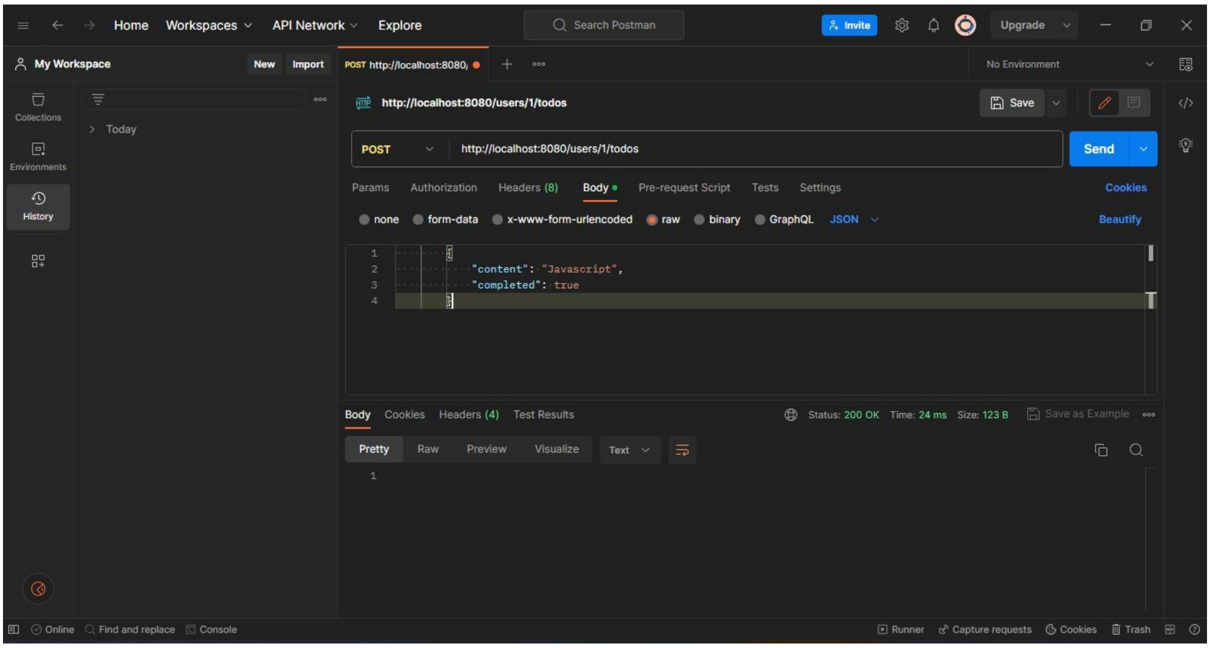
GET http://localhost:8080/users/1

Body: ● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

This request does not have a body

Status: 200 OK Time: 95 ms Size: 327 B

```
{
    "id": 1,
    "username": "Dummy User",
    "password": "password@#",
    "todoList": [
        {
            "id": 1,
            "content": "Springboot",
            "completed": true
        },
        {
            "id": 2,
            "content": "React",
            "completed": false
        }
    ]
}
```



POST http://localhost:8080/users/1/todos

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "content": "Javascript",
    "completed": true
}
```

Status: 200 OK Time: 24 ms Size: 123 B