Step 6: Testing the API

In this step, we'll test the CRUD operations you implemented to make sure they work as expected.

Methods for Testing

- **Unit Testing**
    - You can write unit tests using frameworks like JUnit and Mockito.
    - Create a test package parallel to your **controller** package and annotate your test class with **@SpringBootTest**.
- **Manual Testing**
    - Use tools like Postman to manually send HTTP requests to your API and inspect the responses.
- **Browser**
    - For read-only operations, you can directly access your API via a web browser.

Steps for Manual Testing using Postman

- **Run Your Application**
    - Start your Spring Boot application by clicking on the 'Run' button in IntelliJ.
- **Open Postman**
    - Launch Postman on your computer.
- **Test GET Request**
    - Create a new request in Postman.
    - Set the method to **GET**.
    - Enter the URL as **http://localhost:8080/api/highscores**.
    - Click **Send**.
- **Test POST Request**
    - Create a new request in Postman.
    - Set the method to **POST**.
    - Enter the URL as **http://localhost:8080/api/highscores**.
    - In the **Body** section, select **raw** and **JSON (application/json)**.
    - Add a JSON object for a new high score and click **Send**.

- **Test PUT and DELETE Requests**
    - Similar to **POST**, but you'll use **PUT** and **DELETE** methods and provide an ID in the URL.

Documentation

Update your **README.md**:

markdown

Copy code

```
## Step 6: Testing the API

We tested our CRUD operations using both unit testing frameworks and manual testing tools. For manual testing, we used Postman to send HTTP requests to our API and verified the responses.
```

Commit Changes

- If you wrote unit tests, stage your changes: **git add .**
- Commit: **git commit -m "Added tests for CRUD operations"**

Conclusion

Congratulations, you've successfully tested your CRUD operations! This validates that your API works as expected.