

Creating and Sending GET Requests

Assignment: Implement ApplicationController to Fetch Data from API

Objective

Your task is to implement a Spring Boot REST API controller named ApplicationController that fetches data from an external API using RestTemplate. Your implementation should pass all functional and structural tests.

Provided Files

You are provided with the following files:

- - CreatingAndSendingGetRequestApplication.java: Spring Boot main class.
- - AppConfig.java: Configuration class providing a RestTemplate bean.
- - Blank ApplicationController.java: You need to implement this file.
- - Pre-written test cases (ApiControllerTest) to validate your implementation.

Task Details

Implement the following in ApplicationController.java:

1. Class must be annotated with RestController annotation.
2. Constructor-based injection for RestTemplate must be used.
3. Implement a method named fetchDataFromApi of type public.
4. Method must be annotated with get type with value as /fetchData.
5. Method must accept a request parameter named apiUrl using annotation.
6. Method must return String.
7. Use RestTemplate.getForObject(apiUrl, String.class) to fetch data from the provided API URL and return it.

Note: You can use the endpoint <https://jsonplaceholder.typicode.com/posts/1> in Postman and pass it into the **127.0.0.1/fetchData** URL parameter for testing.

Execution Steps to Follow

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers need to go to the Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal.
3. cd into your backend project folder.
4. To build your project use command:

mvn clean package -Dmaven.test.skip

5. To launch your application, move into the target folder (**cd target**). Run the following command to run the application:

java -jar <your application jar file name>

6. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN. Please use 127.0.0.1 instead of localhost to test rest endpoints.

7. Mandatory: Before final submission run the following command:

mvn test

8. You need to use **CTRL+Shift+B** - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.