API Request and Response Handling

Assignment: Implement AppController to Handle API Request and Response

Objective

Your task is to implement a Spring Boot REST API controller named AppController that sends a GET request to an external API using RestTemplate and handles the response properly, including error handling. Your implementation should pass all functional and structural tests.

Provided Files

1. ApiRequestAndResponseHandlingApplication.java: Spring Boot main class.
2. AppConfig.java: Configuration class providing a RestTemplate bean.
3. Blank AppController.java: You need to implement this file.
4. Pre-written test cases (ApiControllerTest) to validate your implementation.

Task Details

1. Class must be annotated with RestController annotation.
2. Constructor-based injection for RestTemplate must be used.
3. Implement a method named sendApiRequest of type public.
4. Method must be annotated with GET type with value as /sendApiRequest.
5. Method must accept request parameter named apiUrl using annotation.
6. Use RestTemplate.getForEntity(apiUrl, String.class) to send a GET request and fetch data from the provided API URL and return it.
7. Method must return String as "API Response: " + response.
8. Handle HttpClientErrorException for client-side errors (e.g., 404 or 400 errors) and return the status code as "Error: Client-side error - " + e.getStatusCode().
9. Handle general exception:s (e.g., connection issues) and return the exception message as: "Error: " + e.getMessage() .

Note: You can use the endpoint **https://jsonplaceholder.typicode.com/posts/1** in Postman and pass it into the **127.0.0.1/sendApiRequest** 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.**