

API Assignment Questions

1. Explain the following:

A. PUT and PATCH methods

1. PUT Method

- **Purpose:** Replaces the **entire resource** with new data.
- **Behaviour:** If any field is missing in the request body, it is either removed or reset on the server.
- **Idempotent:** Yes — sending the same request multiple times results in the same state.

2. PATCH Method

- **Purpose:** Updates **only specific fields** in the resource.
- **Behaviour:** Does a **partial update**. Fields not included in the request remain unchanged.
- **Idempotent:** Yes — typically considered idempotent, but this depends on implementation.

B. Headers and Cookies

1. Headers:

- Headers are key-value pairs sent in HTTP requests and responses.
- They provide **metadata** (information about the request or response).
- Examples:
 - Content-Type: application/json
 - Authorization: Bearer <token>
 - User-Agent: Mozilla/5.0

2. Cookies:

- Cookies are small pieces of data stored on the client (browser) and sent with each request to the server.
- Used for **session management**, **user tracking**, and **personalization**.
- Example:
Set-Cookie: sessionId=abc123; Path=/; HttpOnly

C. Endpoint and Base URL

Base URL:

- The **common root address** for all API calls.
- Example:
https://api.example.com

Endpoint:

- A specific **path** added to the base URL to access a particular resource.
- Example:

- Base URL: <https://api.example.com>
- Endpoint: `/users/123`
- Full URL: <https://api.example.com/users/123>

D. Query Parameters and Path Parameters

1. Path Parameters:

- Part of the **URL path** used to identify specific resources.
- Example:
GET `/products/45`
 - 45 is a path parameter (product ID)

2. Query Parameters:

- Appended to the URL **after a question mark (?)**.
- Used for **filtering, sorting, or pagination**.
- Example:
GET `/products?category=shoes&sort=price`

E. What are error codes? Explain all series of error codes.

Error codes (or HTTP status codes) are issued by a server in response to a client's request. They indicate the result of the request.

Here are the **5 series** of status codes:

Series	Code Range	Meaning	Examples and Description
1xx	100–199	Informational responses	100 Continue, 101 Switching Protocols
2xx	200–299	Success	200 OK (request successful), 201 Created (resource created)
3xx	300–399	Redirection	301 Moved Permanently, 302 Found, 304 Not Modified
4xx	400–499	Client errors	400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found
5xx	500–599	Server errors	500 Internal Server Error, 502 Bad Gateway, 503 Service Unavailable

2. Create a new collection in Postman named "Sample APIs." Use <https://reqres.in/> and implement GET, POST, PUT, PATCH, and DELETE operations.

https://api.postman.com/collections/42836423-1aa4b90f-867a-4ad5-96d9-e22834f7eac1?access_key=PMAT-01JY6G6H50FZ0P6S9JJNZBBD7R

3. What is the difference between `given()`, `when()`, and `then()` in Rest Assured?

1. given() — Setup / Pre-condition

- Used to set up the request, such as headers, parameters, body, authentication, etc.
- This represents the "Given" step of a BDD test.

2. when() — Action / Trigger

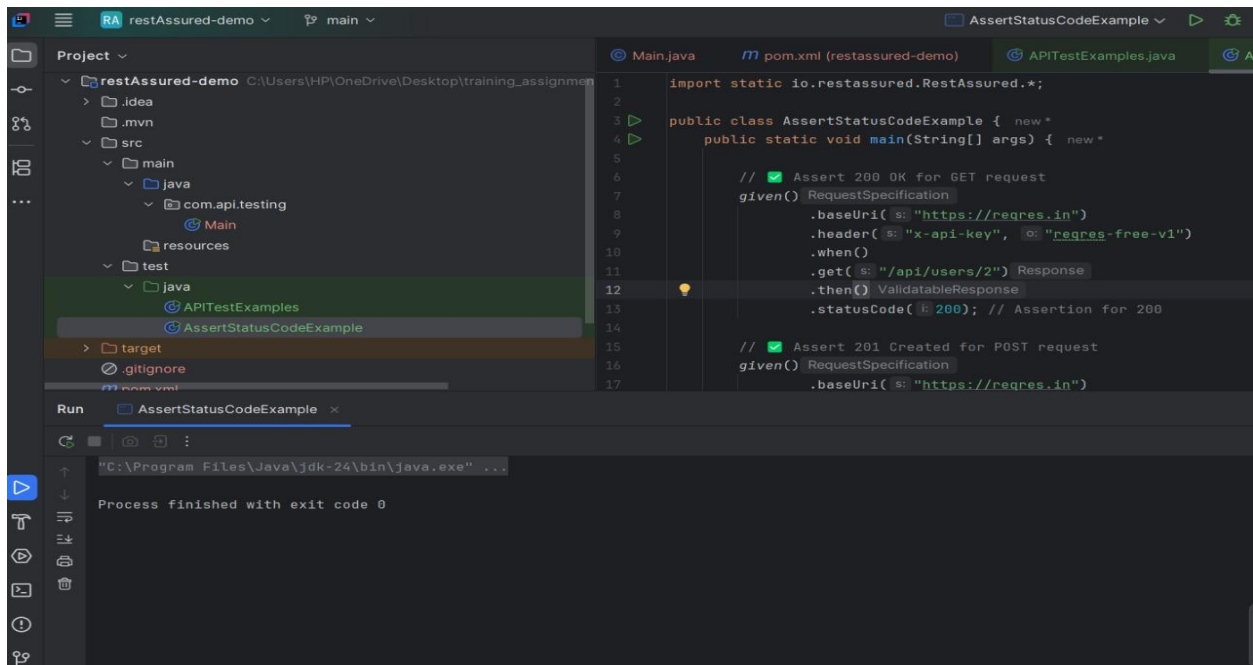
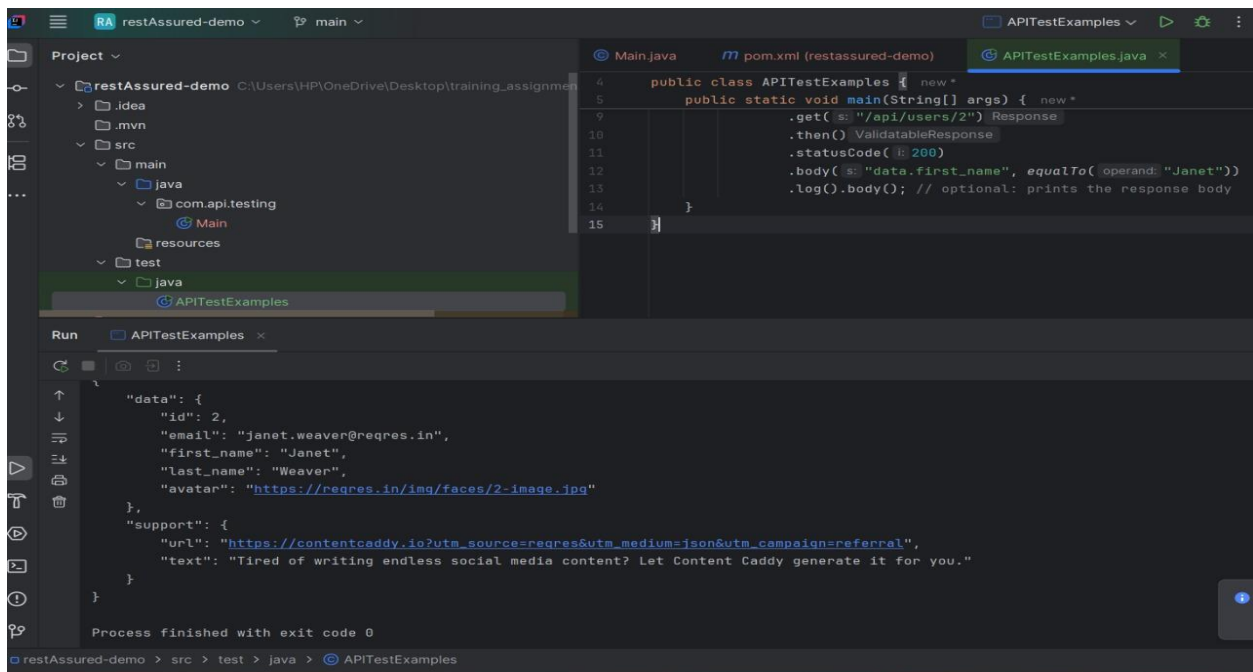
- Used to specify the HTTP method (GET, POST, PUT, DELETE, etc.) and trigger the API request.
- This represents the "When" step in BDD, describing the action being performed.

3. then() — Validation / Assertion

- Used to verify the response received from the server.
- You perform assertions here on status code, response body, headers, etc.
- This represents the "Then" step in BDD, describing the expected outcome.

4. Perform the operations with Rest Assured (use <https://reqres.in/>):

- A. Create GET, POST, PUT, PATCH, and DELETE
- B. Assert for 200/201 response
- C. Verify "first_name" key value from GET response
- D. Extract "email" from GET request and add it to the subsequent POST request payload and hit the request
- E. Write a function to verify presence of Message in response body



restAssured-demo main

Project

- .idea
- .mvn
- src
 - main
 - java
 - com.apitesting
 - Main

Run ExtractEmailAndPostExample

Examples.java

```
1 import io.restassured.response.Response;
2
3 import static io.restassured.RestAssured.*;
4 import static org.hamcrest.Matchers.*;
5
6 public class ExtractEmailAndPostExample {
7     public static void main(String[] args) {
```

Extracted Email: janet.weaver@reqres.in

Request method: POST

Request URI: <https://reqres.in/api/users>

Proxy: <none>

Request params: <none>

Query params: <none>

Form params: <none>

Path params: <none>

Headers: x-api-key=reqres-free-v1

Accept= */*

Content-Type=application/json

Cookies: <none>

Multiparts: <none>

Body:

```
{
  "name": "Shtakshi",
  "email": "janet.weaver@reqres.in"
}
```

HTTP/1.1 201 Created

Date: Sun, 22 Jun 2025 12:30:55 GMT

Content-Type: application/json; charset=utf-8

restAssured-demo > src > test > java > ExtractEmailAndPostExample

