

API Assignment Questions

1. Explain the following:

- A. PUT and PATCH methods
 - B. Headers and Cookies
 - C. Endpoint and Base URL
 - D. Query Parameters and Path Parameters
 - E. What are error codes? Explain all series of error codes.
2. Create a new collection in Postman named "Sample APIs." Use <https://reqres.in/> and implement GET, POST, PUT, PATCH, and DELETE operations.
3. What is the difference between `given()`, `when()`, and `then()` in Rest Assured?

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

Answers

1

A. PUT and PATCH Methods:

- **PUT:** Used to update or replace an entire resource. It is idempotent, meaning multiple identical requests will yield the same result. Example: Updating a user's details with a new object.
- **PATCH:** Used to apply partial updates to a resource. Not necessarily idempotent. Example: Updating only the user's age.

B. Headers and Cookies:

- **Headers:** Metadata sent with requests or responses, such as Authorization, Content-Type, or Accept. Used to pass additional information between client and server. Example: { "Content-Type": "application/json" }.
- **Cookies:** Data stored on the client side, sent with every HTTP request to the server. Used for session management, personalization, or tracking. Example: Set-Cookie: sessionId=abc123; Path=/; HttpOnly.

C. Endpoint and Base URL:

- **Base URL:** The root address of an API. Example: https://api.krishna.com/.
- **Endpoint:** A specific path added to the Base URL to access a particular resource or functionality. Example: /details/1.

D. Query Parameters and Path Parameters:

- **Query Parameters:** Appended to the URL after ?. Used to filter or modify the request. Example: https://api.krishna.com/marks?role=teacher.
- **Path Parameters:** Embedded in the URL path and used to identify specific resources. Example: https://api.krishna.com/marks/{id}.

E. Error Codes:

- Error codes indicate the result of an HTTP request.

1xx (Informational): Request received, continuing process. Example: 100 Continue.

2xx (Success): Request successfully processed. Example: 200 OK, 201 Created.

3xx (Redirection): Further action needs to be taken. Example: 301 Moved Permanently, 302 Found.

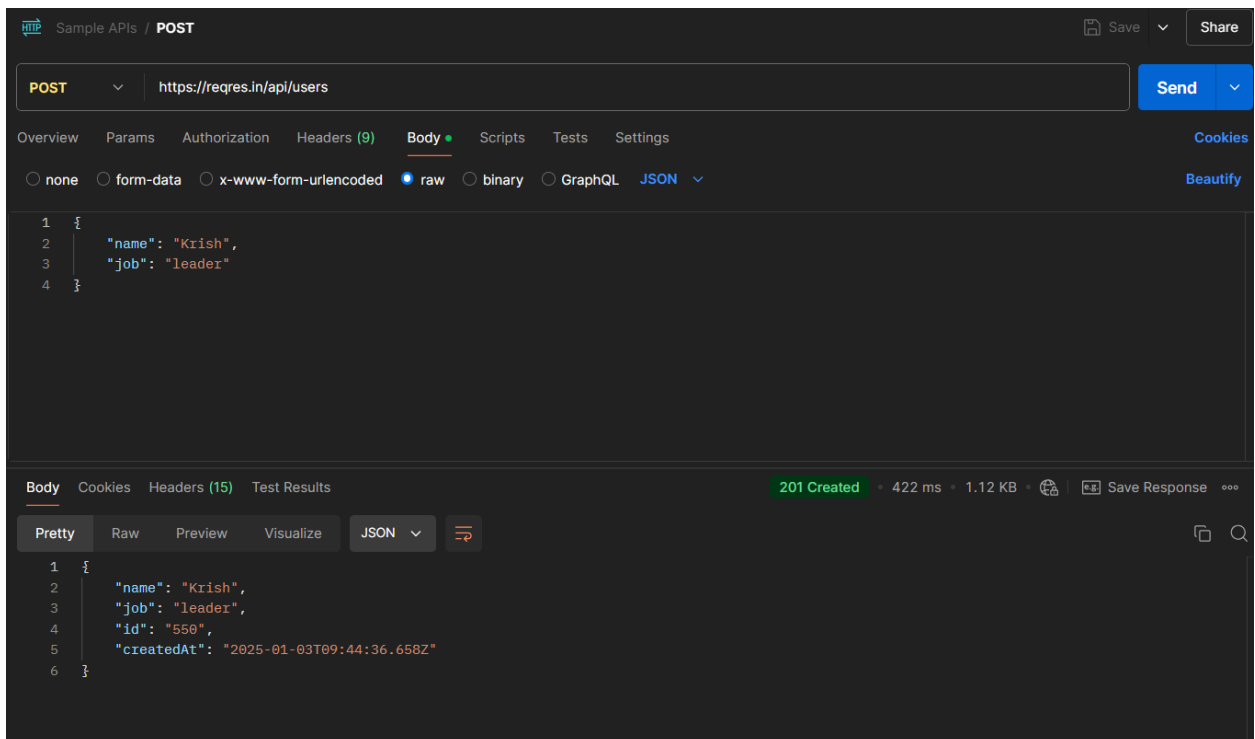
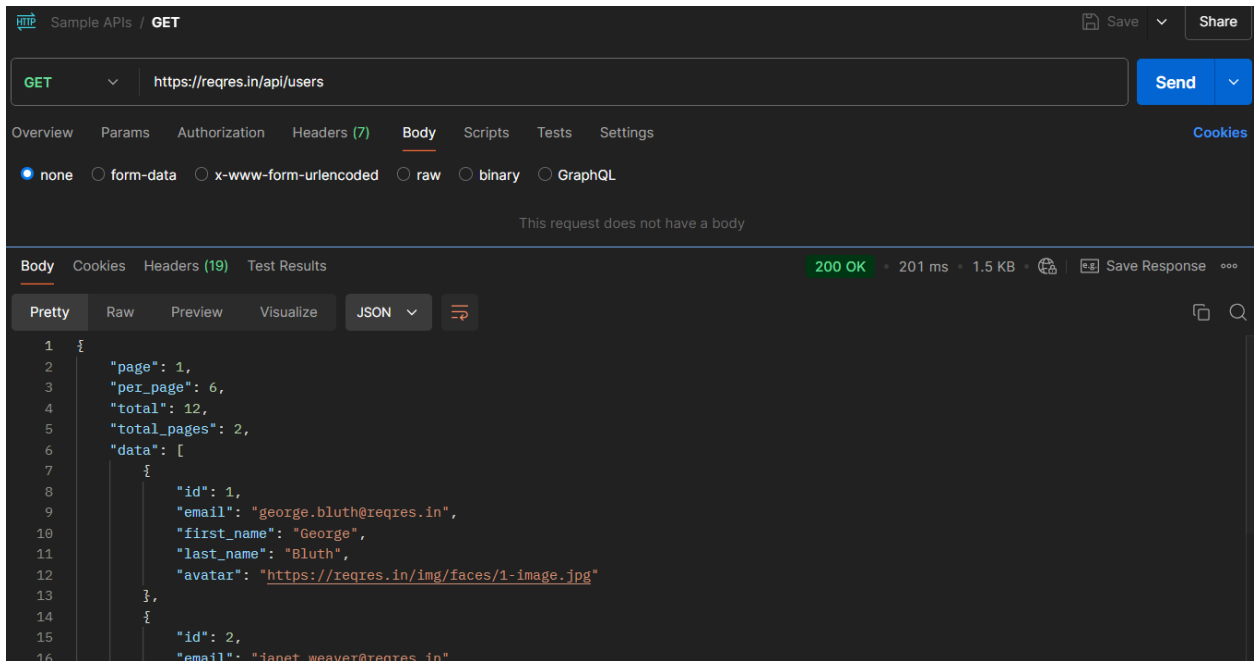
4xx (Client Side Errors): Issues with the client request. Example: 400 Bad Request, 401 Unauthorized, 404 Not Found.

5xx (Server Side Errors): Issues on the server side. Example: 500 Internal Server Error, 503 Service Unavailable.

2

Postman Collection Link : [link](#)

Snapshots:



Sample APIs / PUT

PUT

https://reqres.in/api/users/2

Send

Overview

Params

Authorization

Headers (9)

Body

Scripts

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "name": "Krish",
3   "job": "zion resident"
4 }
```

Body

Cookies

Headers (17)

Test Results

200 OK

515 ms

1.17 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "name": "Krish",
3   "job": "zion resident",
4   "updatedAt": "2025-01-03T09:44:42.991Z"
5 }
```

Sample APIs / PATCH

PATCH

https://reqres.in/api/users/2

Send

Overview

Params

Authorization

Headers (9)

Body

Scripts

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "name": "Krishna",
3   "job": "zion resident"
4 }
```

Body

Cookies

Headers (16)

Test Results

200 OK

391 ms

1.14 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "name": "Krishna",
3   "job": "zion resident",
4   "updatedAt": "2025-01-03T09:44:48.422Z"
5 }
```

Sample APIs / DELETE

DELETE

https://reqres.in/api/users/2

Send

Overview

Params

Authorization

Headers (7)

Body

Scripts

Tests

Settings

Cookies

DELETE

View complete documentation →

https://reqres.in/api/users/2

Add request description...

Code Snippet

Body

Cookies

Headers (14)

Test Results

204 No Content

299 ms

1017 B

Save Response

Pretty

Raw

Preview

Visualize

Text

```
1
```

3

- `given()`: Used to define the preconditions or setup for the request. Example: Setting headers, query parameters, or request body.
- `when()`: Defines the action to be performed, such as making a GET, POST, PUT, or DELETE request.
- `then()`: Used to specify the expected outcomes or assertions for the response.

Example in Rest Assured:

```
given()
  .header("Content-Type", "application/json")
  .body("{\"name\": \"Krishna\", \"job\": \"Software Developer\"}")
.when()
  .post("https://api.krishna.in /users")
.then()
  .statusCode(201)
  .body("name", equalTo("Krishna"));
```

4

```
package api;

import io.restassured.RestAssured;
import io.restassured.http.ContentType;
import io.restassured.response.Response;

import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.*;

public class RestAssuredGeminiAPI {

    public static void main(String[] args) {
        RestAssured.baseURI = "https://reqres.in/api";

        // GET
        Response getResponse = given()
            .when()
            .get("/users/2")
            .then()
            .statusCode(200) //200
            .extract().response();

        //verify first_name
        String firstName = getResponse.jsonPath().getString("data.first_name");
        System.out.println("First Name: " + firstName);

        // email from get res
        String email = getResponse.jsonPath().getString("data.email");
        System.out.println("Email: " + email);

        // post req on email
        String postPayload = "{ \"name\": \"Krishna\", \"job\": \"Developer\", \"email\": \"\" +
email + "\" }";
        Response postResponse = given()
```

```

        .contentType(ContentType.JSON)
        .body(postPayload)
        .when()
        .post("/users")
        .then()
        .statusCode(201) //201 created
        .extract().response();

System.out.println("POST Response: " + postResponse.asString());

// PUT
String putPayload = "{ \"name\": \"Krishna\", \"job\": \"President\" }";
given()
    .contentType(ContentType.JSON)
    .body(putPayload)
    .when()
    .put("/users/2")
    .then()
    .statusCode(200); //200

// PATCH
String patchPayload = "{ \"name\": \"Krishna\" }";
given()
    .contentType(ContentType.JSON)
    .body(patchPayload)
    .when()
    .patch("/users/2")
    .then()
    .statusCode(200); // 200

// DELETE
given()
    .when()
    .delete("/users/2")
    .then()
    .statusCode(204); // 204 no content

//verify message in body
boolean isMessagePresent = isKeyPresent(postResponse, "message");
System.out.println("Is 'message' key present in res body? " + isMessagePresent);
}

// Function to verify presence of a key in response body
public static boolean isKeyPresent(Response response, String key) {
    try {
        response.jsonPath().get(key);
        return true;
    } catch (Exception e) {
        return false;
    }
}
}
}

```

<terminated> RestAssuredGeminiAPI [Java Application] C:\Users\dell\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full\jre\bin\java.exe

First Name: Janet

Email: janet.weaver@reqres.in

POST Response: {"name":"Krishna","job":"Developer","email":"janet.weaver@reqres.in","id":"1234567890"};

Is 'message' key present in res body? true