

API Assignment Questions

1. Explain the following:

A. PUT and PATCH methods

- **PUT**

- Replaces the **entire resource** at the specified URI.
- It is **idempotent**: calling it multiple times with the same data will not change the result.
- Example: `PUT /users/1` with updated user details replaces the entire user object.

- **PATCH**

- Updates **part of a resource**.
- Not necessarily idempotent.
- Example: `PATCH /users/1` with `{ "email": "new@example.com" }` only updates the email.

B. Headers and Cookies

- **Headers**

- Key-value pairs sent in HTTP requests and responses.
- Used to convey metadata such as:
 - `Content-Type`: type of the request/response body
 - `Authorization`: credentials or tokens
 - `Accept`: expected response format

- **Cookies**

- Small data stored on the client by the server.
- Used for session management, tracking, personalization.
- Sent automatically with each request to the same domain.

C. Endpoint and Base URL

- **Base URL**

- The **root address** of the API.
- Example: `https://api.example.com`

- **Endpoint**

- The **specific path** added to the base URL to access a resource.
- Example: `https://api.example.com/users/1` — here, `/users/1` is the endpoint.

D. Query Parameters and Path Parameters

- **Path Parameters**

- Part of the **URL path**.
- Used to **identify a specific resource**.
- Example: `/users/123` → 123 is a path parameter (user ID).

- **Query Parameters**

- Part of the **URL after the ? symbol**.
- Used for **filtering, sorting, pagination**, etc.
- Example: `/users?page=2&limit=10` → page and limit are query parameters.

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

HTTP status codes are 3-digit numbers returned in HTTP responses to indicate result of a request.

Code Series	Meaning	Examples
1xx	Informational	100 Continue, 101 Switching Protocols
2xx	Success	200 OK, 201 Created, 204 No Content
3xx	Redirection	301 Moved Permanently, 302 Found
4xx	Client Errors	400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found
5xx	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/46085162-0d0f8679-46bc-4c18-a723-6e21e8751c4a?access_key=PMAT-01JYE5E3DNCNTB4W1B8SB4ZSSR

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

1. `given()` — Setup / Precondition

- Used to define the **request specification**.
- This includes:
 - Request headers
 - Authentication
 - Parameters (query/path/form)
 - Request body (JSON, XML, etc.)

Example:

```
java
Copy code
given()
    .header("Content-Type", "application/json")
    .body("{ \"name\": \"John\" }")
```

2. `when()` — Action

- Specifies **what action to perform**, usually the HTTP method (GET, POST, PUT, DELETE).
- Triggers the API call.

Example:

```
java
Copy code
.when()
    .post("/api/users")
```

3. `then()` — Validation / Assertion

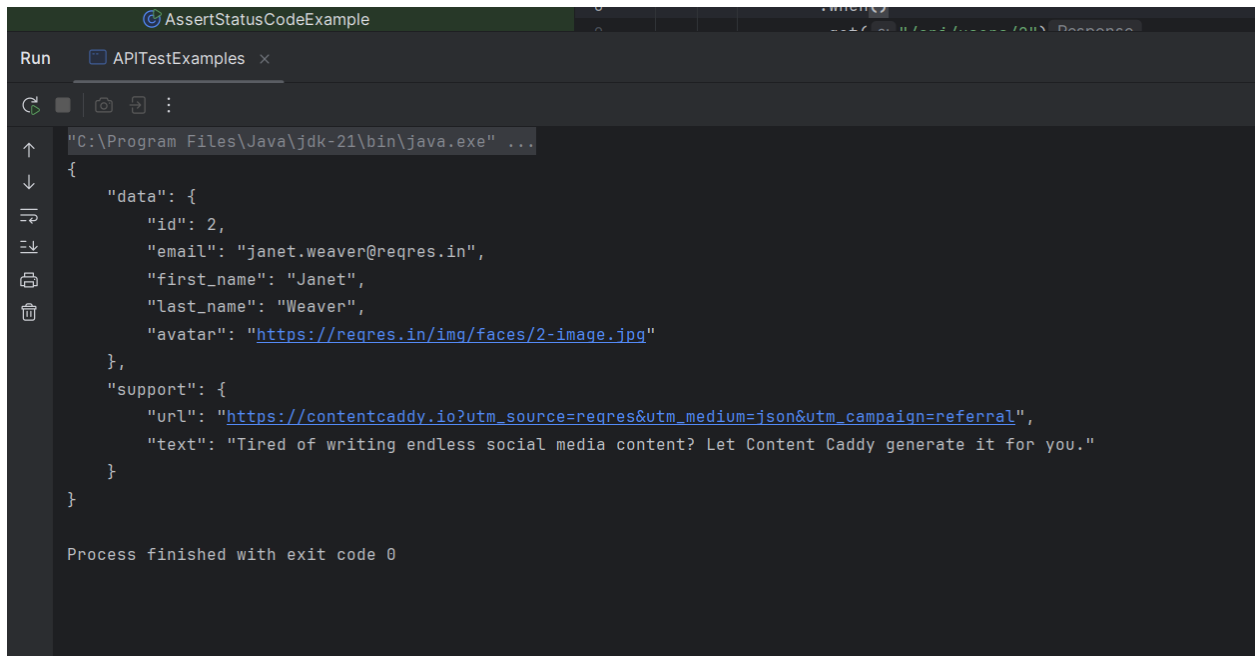
- Used to **verify** the response.
- Common assertions include:
 - Status code
 - Response body values
 - Headers

Example:

```
java
Copy code
.when()
    .post("/api/users")
```

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

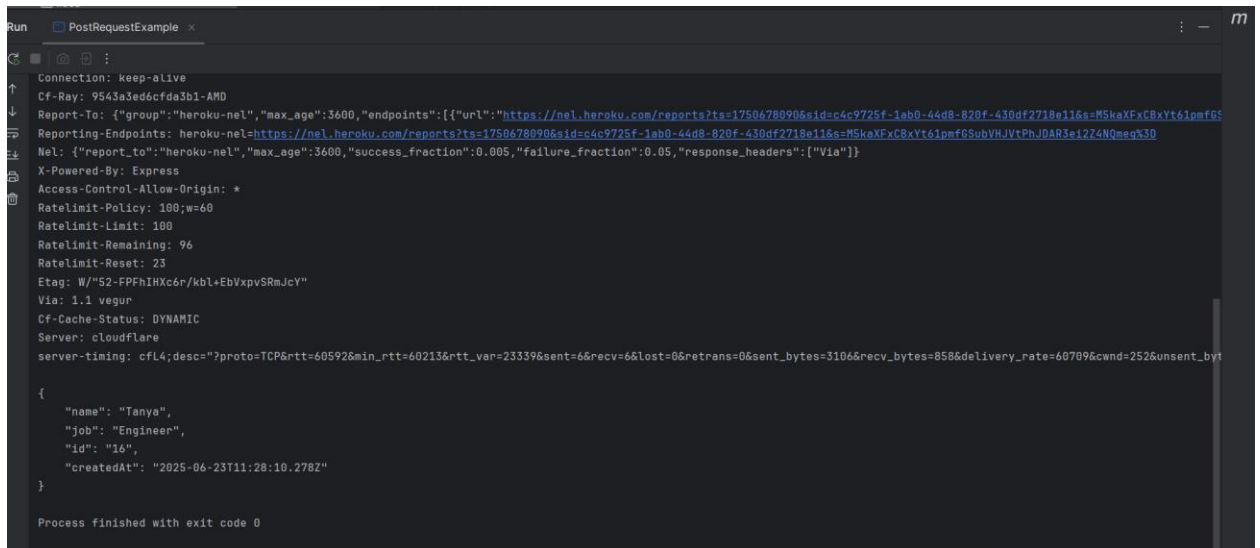
A. Create GET, POST, PUT, PATCH, and DELETE
GET:



The screenshot shows an IDE window titled "AssertStatusCodeExample". The "Run" tab is active, displaying a JSON response from a GET request. The response is a JSON object with a "data" field containing user information and a "support" field containing a URL and a message. The process finished with exit code 0.

```
"C:\Program Files\Java\jdk-21\bin\java.exe" ...  
{  
  "data": {  
    "id": 2,  
    "email": "janet.weaver@reqres.in",  
    "first_name": "Janet",  
    "last_name": "Weaver",  
    "avatar": "https://reqres.in/img/faces/2-image.jpg"  
  },  
  "support": {  
    "url": "https://contentcaddy.io?utm_source=reqres&utm_medium=json&utm_campaign=referral",  
    "text": "Tired of writing endless social media content? Let Content Caddy generate it for you."  
  }  
}  
  
Process finished with exit code 0
```

POST:



The screenshot shows an IDE window titled "PostRequestExample". The "Run" tab is active, displaying a JSON response from a POST request. The response is a JSON object with a "name" field, a "job" field, an "id" field, and a "createdAt" field. The process finished with exit code 0.

```
Connection: keep-alive  
Cf-Ray: 9543a3ed6cfda3b1-AMD  
Report-To: {"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.com/reports?ts=1750678090&sid=c4c9725f-1ab0-44d8-820f-430df2718e11&s=M5kaXFc8xYt61pmfG5"}]}  
Reporting-Endpoints: heroku-nel=https://nel.heroku.com/reports?ts=1750678090&sid=c4c9725f-1ab0-44d8-820f-430df2718e11&s=M5kaXFc8xYt61pmfG5SubVHJVtPhJdARiE1274NQmeg530  
Nel: {"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}  
X-Powered-By: Express  
Access-Control-Allow-Origin: *  
Ratelimit-Policy: 100;w=60  
Ratelimit-Limit: 100  
Ratelimit-Remaining: 96  
Ratelimit-Reset: 23  
Etag: W/"52-FPFhIHxc6r/kbl+EbVxpvSRmJcY"  
Via: 1.1 vegur  
Cf-Cache-Status: DYNAMIC  
Server: cloudflare  
server-timing: cfL4;desc="?proto=TCP&rtt=60592&min_rtt=60213&rtt_var=23339&sent=6&recv=6&lost=0&retrans=0&sent_bytes=3106&recv_bytes=858&delivery_rate=60709&cwnd=252&unsent_byt  
  
{  
  "name": "Tanya",  
  "job": "Engineer",  
  "id": "16",  
  "createdAt": "2025-06-23T11:28:10.278Z"  
}  
  
Process finished with exit code 0
```

PUT:

```
Run PutRequestExample x
Content-Encoding: gzip
Report-To: {"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.com/reports?ts=1750678180&sid=c4c9725f-1ab0-44d8-820f-430df2718e11&s=71u7HnUakIQG8H3bgo7"}]}
Reporting-Endpoints: heroku-nel=https://nel.heroku.com/reports?ts=1750678180&sid=c4c9725f-1ab0-44d8-820f-430df2718e11&s=71u7HnUakIQG8H3bgo7w691QJP10%28wuw9Dpmvbwrt8%3D
Nel: {"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
X-Powered-By: Express
Access-Control-Allow-Origin: *
Ratelimit-Policy: 100;w=60
Ratelimit-Limit: 100
Ratelimit-Remaining: 96
Ratelimit-Reset: 25
Etag: W/"4f-tx8JScjMv2J/928HAbPLm/z61WM"
Via: 1.1 vegur
Cf-Cache-Status: DYNAMIC
Vary: Accept-Encoding
Server: cloudflare
Cf-Ray: 9543a61f590f7935-AMD
server-timing: cfL4;desc="?proto=TCP&rtt=60227&min_rtt=59935&rtt_var=23060&sent=7&recv=7&lost=0&retrans=0&sent_bytes=3107&recv_bytes=866&delivery_rate=61662&wnd=252&unsent_byt

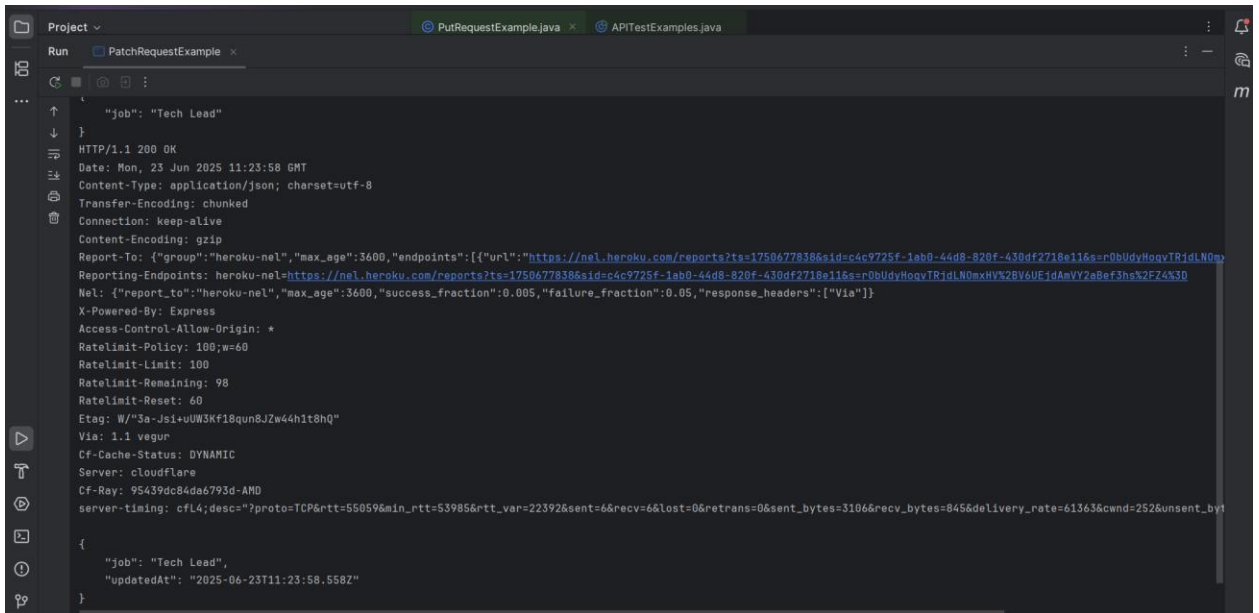
{
  "name": "Tanya",
  "job": "Senior Engineer",
  "updatedAt": "2025-06-23T11:29:40.231Z"
}

Process finished with exit code 0
```

DELETE:

```
Run DeleteRequestExample x
Proxy: <none>
Request params: <none>
Query params: <none>
Form params: <none>
Path params: <none>
Headers: x-api-key=reqres-free-v1
Accept: */*
Cookies: <none>
Multiparts: <none>
Body: <none>
HTTP/1.1 204 No Content
Date: Mon, 23 Jun 2025 11:26:06 GMT
Connection: keep-alive
Cf-Ray: 9543a0e408407943-AMD
Report-To: {"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.com/reports?ts=1750677965&sid=c4c9725f-1ab0-44d8-820f-430df2718e11&s=Kep8k3yVqfwJzi8FMU"}]}
Reporting-Endpoints: heroku-nel=https://nel.heroku.com/reports?ts=1750677965&sid=c4c9725f-1ab0-44d8-820f-430df2718e11&s=Kep8k3yVqfwJzi8FMUptoyzCXMzelyvEkN3a56elkd1%3D
Nel: {"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
X-Powered-By: Express
Access-Control-Allow-Origin: *
Ratelimit-Policy: 100;w=60
Ratelimit-Limit: 100
Ratelimit-Remaining: 98
Ratelimit-Reset: 60
Etag: W/"2-vy6p6PvFo4RvsFtPoIWeCRyIC8"
Via: 1.1 vegur
Cf-Cache-Status: DYNAMIC
Server: cloudflare
server-timing: cfL4;desc="?proto=TCP&rtt=60571&min_rtt=60277&rtt_var=23192&sent=6&recv=6&lost=0&retrans=0&sent_bytes=3106&recv_bytes=772&delivery_rate=61312&wnd=252&unsent_byt
```

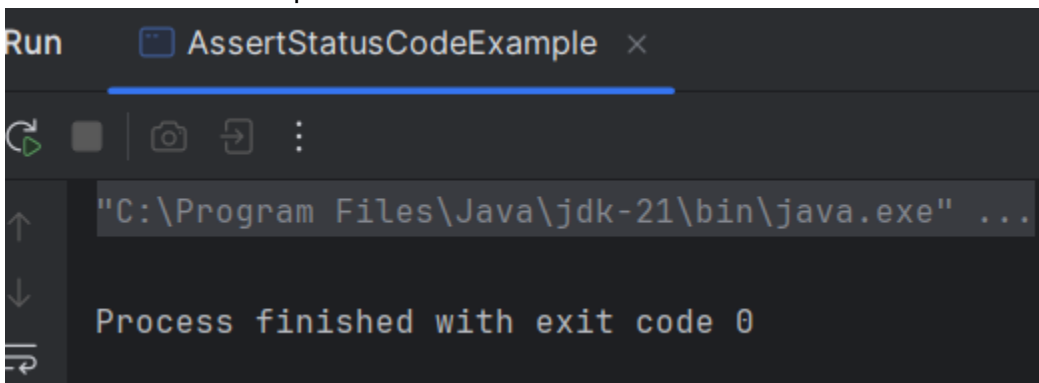
PATCH:



```
Project ▾ PutRequestExample.java APIExamples.java
Run PatchRequestExample ▾
  "job": "Tech Lead"
}
HTTP/1.1 200 OK
Date: Mon, 23 Jun 2025 11:23:58 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Content-Encoding: gzip
Report-To: {"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.com/reports?ts=1750677838&sid=c4c9725f-1eb0-44d8-820f-430df2718e11&sr=0bUdyHoqvTRIdLN0m..."}]}
Reporting-Endpoints: heroku-nel=https://nel.heroku.com/reports?ts=1750677838&sid=c4c9725f-1eb0-44d8-820f-430df2718e11&sr=0bUdyHoqvTRIdLN0m...
Nel: {"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
X-Powered-By: Express
Access-Control-Allow-Origin: *
Ratelimit-Policy: 100;w=60
Ratelimit-Limit: 100
Ratelimit-Remaining: 98
Ratelimit-Reset: 60
Etag: W/"3a-JsI+uUM3Kf18qun8JZw44h1t8hQ"
Via: 1.1 vegur
Cf-Cache-Status: DYNAMIC
Server: cloudflare
Cf-Ray: 95439dc84da6793d-AMD
server-timing: cfL4;desc=?proto=TCP&rtt=55059&min_rtt=53985&rtt_var=22392&sent=66&recv=66&lost=0&retrans=0&sent_bytes=3106&recv_bytes=845&delivery_rate=61363&cwnd=252&unsent_byt...

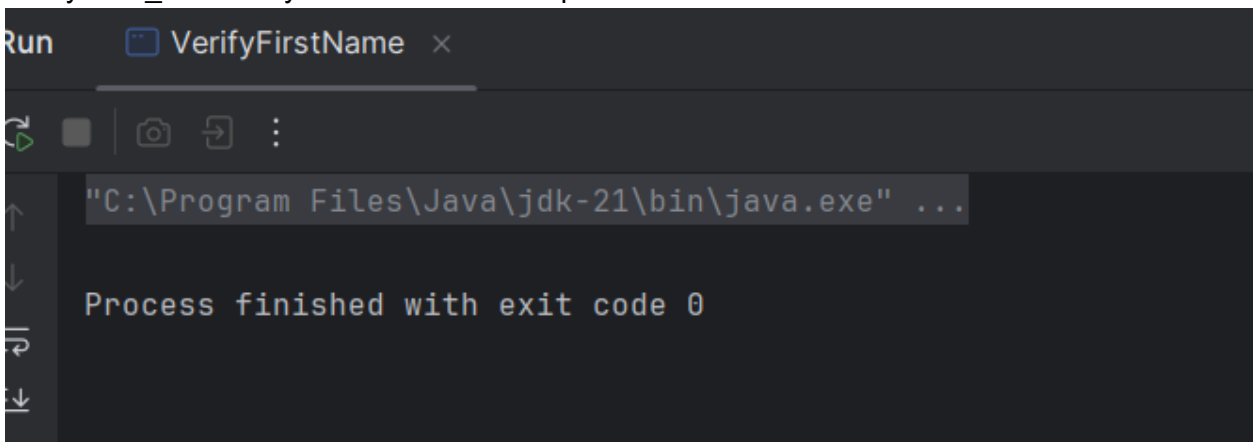
{
  "job": "Tech Lead",
  "updatedAt": "2025-06-23T11:23:58.558Z"
}
```

B. Assert for 200/201 response



```
Run AssertStatusCodeExample ▾
  "C:\Program Files\Java\jdk-21\bin\java.exe" ...
Process finished with exit code 0
```

C. Verify "first_name" key value from GET response



```
Run VerifyFirstName ▾
  "C:\Program Files\Java\jdk-21\bin\java.exe" ...
Process finished with exit code 0
```

D. Extract "email" from GET request and add it to the subsequent POST request payload and hit the request

```
Run ExtractEmailAndPostExample x
Reporting-Endpoints: heroku-nel=https://nel.heroku.com/reports?ts=1750678373&sid=c4c9725f-1ab0-44d8-820f-430df2718e11&s=lf51te1t7288revvve1CXRkb%2F0XhewTvUP%2F1m796Z618%30
Nel: {"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
X-Powered-By: Express
Access-Control-Allow-Origin: *
Ratelimit-Policy: 100;w=60
Ratelimit-Limit: 100
Ratelimit-Remaining: 98
Ratelimit-Reset: 60
Etag: W/"63-1mY8n7MbsQBslKkTcZvET4DmCBQ"
Via: 1.1 vegur
Cf-Cache-Status: DYNAMIC
Server: cloudflare
server-timing: cfL4;desc="?proto=TCP&rtt=147907&min_rtt=62679&rtt_var=80584&sent=5&recv=66&lost=0&retrans=0&sent_bytes=3105&recv_bytes=8746&delivery_rate=61264&cwnd=252&unsent_b
{
  "name": "Tanya",
  "email": "janet.weaver@regres.in",
  "id": "439",
  "createdAt": "2025-06-23T11:32:53.661Z"
}
Process finished with exit code 0
```

E. Write a function to verify presence of Message in response body

```
Run MessagePresenceChecker x
"C:\Program Files\Java\jdk-21\bin\java.exe" ...
Is 'message' key present in response? false
Process finished with exit code 0
```