REST API Interview Questions – Numbered Version

Q1. What is a REST API?

A: A REST API (Representational State Transfer) is an architectural style for building scalable web services. It uses standard HTTP methods such as GET, POST, PUT, DELETE and treats every piece of data as a resource. REST APIs follow principles such as statelessness, uniform interface, client-server architecture, and cacheability, making them lightweight and easy for integration between client and server applications.

```
import requests
response = requests.post("https://api.example.com/login", json={"user": "mahesh"}
print(response.status_code)
```

Q2. What does statelessness mean in REST?

A: Statelessness means the server does not store any client session information. Every API request must contain all information required to process it. This improves scalability because servers can handle each request independently.

```
requests.post("https://api.example.com/order", json={"id": 10})
```

Q3. Explain POST request with a detailed example.

A: POST is used to create a new resource on the server. When we send a POST request, the server processes the data and returns a new resource ID. It is not idempotent — sending twice may create duplicates.

```
requests.post("https://api.example.com/users", json={"name": "Uma"})
```

Q4. What is PUT and when do we use it?

A: PUT replaces an entire resource. It is idempotent — sending same request twice results in same state. Use when updating full record.

```
requests.put("https://api.example.com/user/1", json={"name": "Mahesh", "age": 25}
```

Q5. What is PATCH and how is it different from PUT?

A: PATCH updates only selected fields of a resource, unlike PUT which replaces the whole resource. PATCH is used for partial updates.

```
requests.patch("https://api.example.com/user/1", json={"age": 27})
```

Q6. What are API Headers and why are they important?

A: Headers contain metadata like content type, authentication tokens, cache instructions. They guide server behavior and security.

```
headers = {"Content-Type": "application/json"}
```

Q7. What is JSON and why is it preferred in REST APIs?

A: JSON is a lightweight, language-independent format used for data exchange. It is easy to read, parse, and widely supported.

```
{ "id": 1, "name": "Mahesh" }
```

Q8. What are Query Parameters in API?

A: Used for filtering, sorting, searching and pagination. They are optional and appear after '?' in URL.

```
/users?limit=10&role=admin
```

Q9. What are Path Parameters?

A: Path parameters uniquely identify a resource and are mandatory in URL.

```
/users/101
```

Q10. Explain Authentication vs Authorization with real-time usage.

A: Authentication verifies identity (login). Authorization checks permissions (what user can access).

```
headers = {"Authorization": "Bearer TOKEN"}
```

Q11. Explain different types of HTTP Status Codes.

A: HTTP status codes indicate API request outcome: 2xx success, 4xx client error, 5xx server error.

```
response.status_code
```

Q12. What are Idempotent Methods in REST?

A: Idempotent methods (GET, PUT, DELETE) produce the same result even when called multiple times.

```
requests.delete("https://api.example.com/user/11")
```

Q13. What is Rate Limiting?

A: Rate limiting restricts number of API calls allowed per time window to prevent abuse and server overload.

```
429 Too Many Requests
```

Q14. What is API Response Time?

A: It is total time taken by server to process and respond to a request. It impacts performance and UX.

```
response.elapsed.total_seconds()
```

Q15. What is Postman and why is it used?

A: Postman is a tool for developing, testing, automating, and documenting REST APIs.

```
pm.test("Status is 200", function(){ pm.response.to.have.status(200); });
```

Q16. What is Swagger / OpenAPI?

A: Swagger is used to document, design, and consume APIs via the interactive UI.

```
openapi: 3.0.0
```

Q17. What is API Contract Testing?

A: Ensures API follows a predefined schema defining required fields and types. Prevents breaking changes.

```
assert "id" in response.json()
```

Q18. What is Token-Based Authentication?

A: Uses JWT or tokens instead of sessions; token is included in headers for each call.

```
Authorization: Bearer <JWT_TOKEN>
```

Q19. What is Caching in REST APIs?

A: Caching stores pre-fetched responses to reduce repeated server calls and improve performance.

```
Cache-Control: max-age=3600
```

Q20. What validations are performed during API testing?

A: API testing validates status codes, headers, JSON schema, data accuracy, response time, and business rules.

```
assert response.headers["Content-Type"] == "application/json"
```