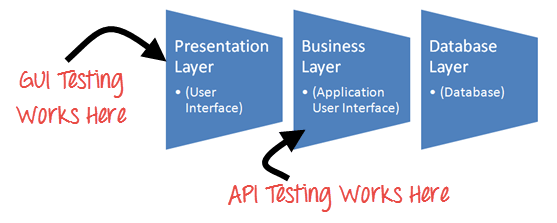
# API & Postman Concepts with Examples

## 1. What is API

API (Application Programming Interface) allows communication between software components.  
Example: Google Maps API lets other apps show maps and get routes.



## 2. Advantages

• Faster development  
• Code reusability  
• Platform independence  
• Easy integration

We can do diff tastings like Validation Testing , 2. Functional Testing , 3. UI testing , 4. Load testing , 5. Runtime/ Error Detection ,6. Security testing , 7. Penetration testing , 8. Fuzz testing , 9. Interoperability and WS Compliance testing .

## 3. HTTP Methods

GET – Fetch data (e.g., GET /users)  
POST – Create data (e.g., POST /users with body {"name": "John"})  
PUT – Update entire data (e.g., PUT /users/1)  
PATCH – Update partial data (e.g., PATCH /users/1 with {"email": "a@test.com"})  
DELETE – Remove data (e.g., DELETE /users/1)

## 4. Differences between API and Web Service

## All Web services are APIs but not all APIs are Web services.

## Web services might not contain all the specifications and cannot perform all the tasks that APIs would perform.

## A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication whereas API may be exposed to in multiple ways.

## A Web service always needs a network to operate while APIs don’t need a network for operation.

## 5. HTTP vs HTTPS

## The HTTP protocol stands for Hypertext Transfer Protocol, whereas the HTTPS stands for Hypertext Transfer Protocol Secure. Security The HTTP protocol is not secure protocol as it does not contain SSL (*Secure Sockets Layer*), which means that the data can be stolen when the data is transmitted from the client to the server.

## 6. REST API

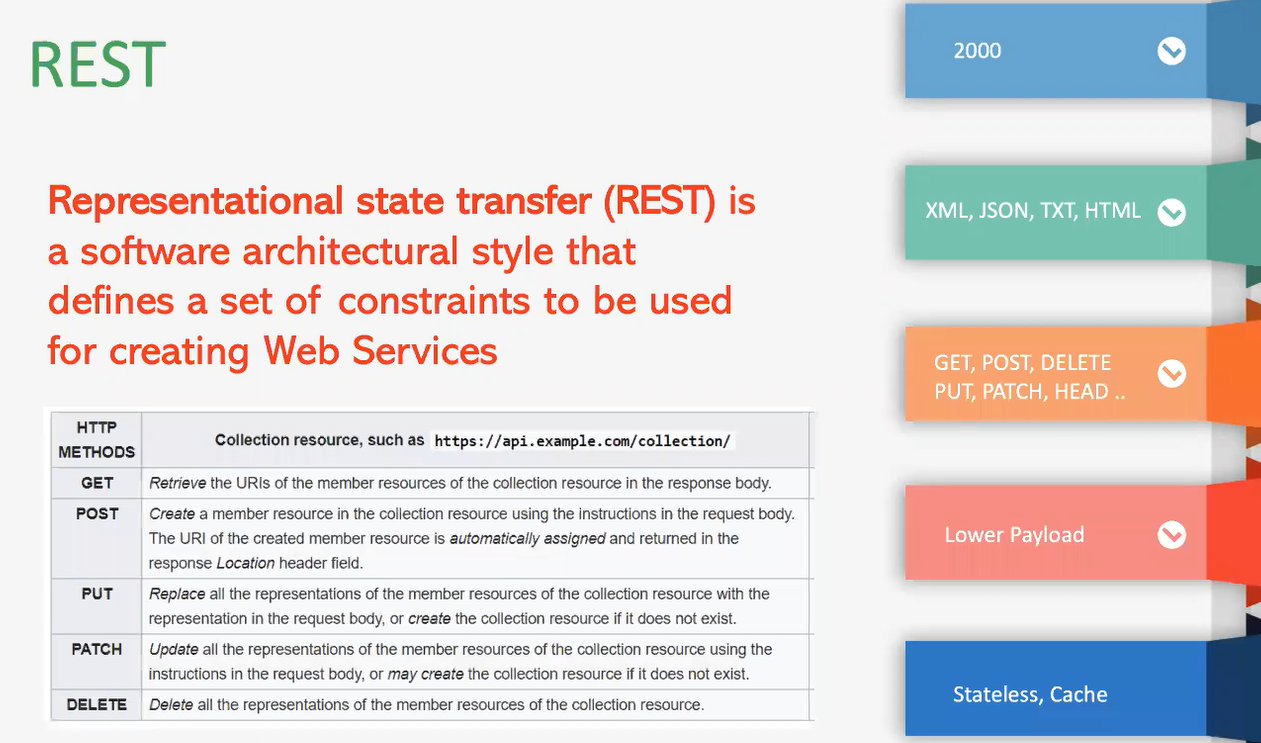
REST (Representational State Transfer) uses HTTP for operations.  
Example: GET /books/1 returns book details.

You search for something, and you get a list of results back from the service you're requesting from. ... The developer creates the API on the **server** and allows the **client** to talk to it. REST determines how the API looks like. It stands for “Representational State Transfer “.

## 7. SOAP

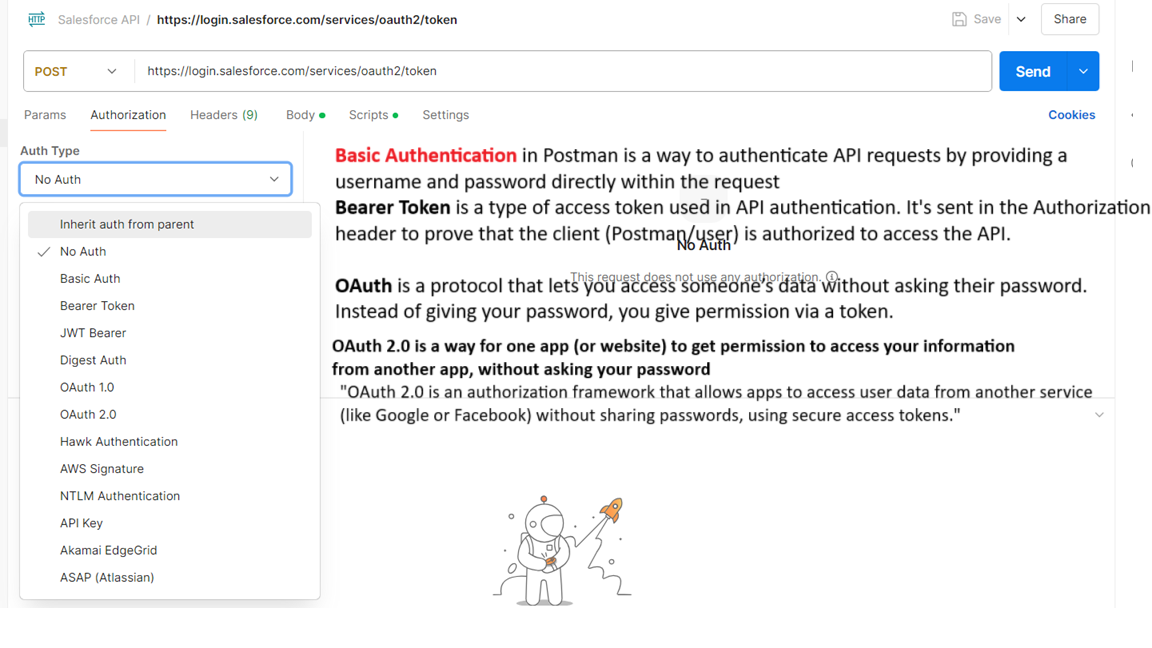
## 8. REST API

Lightweight, stateless, uses JSON, scalable.



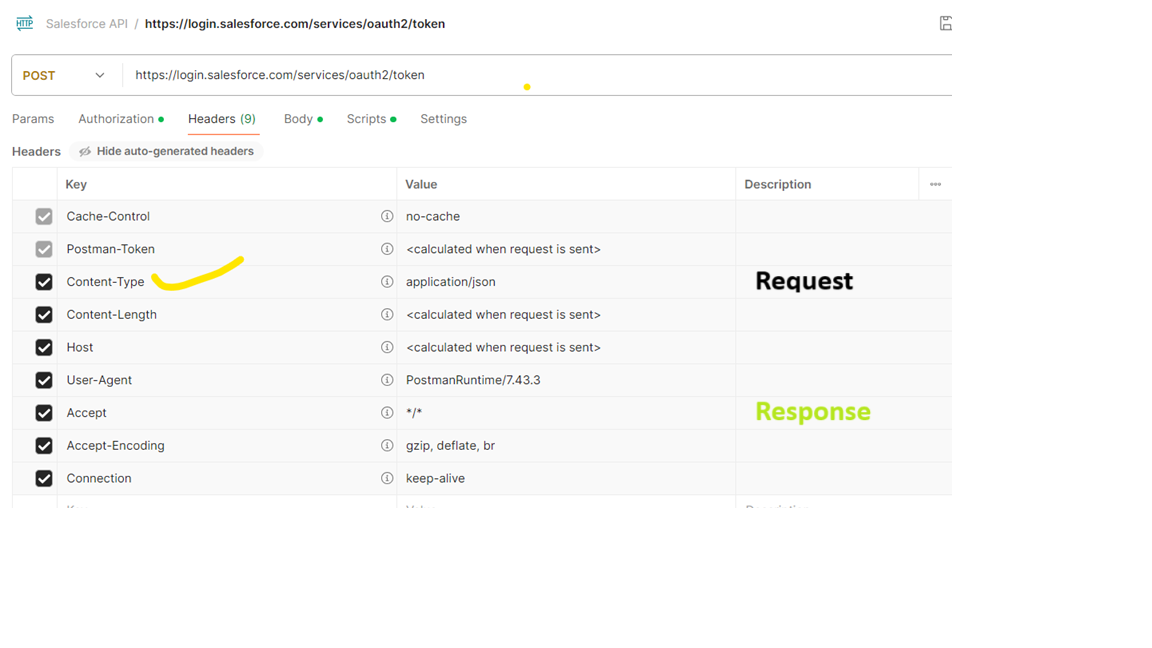
## 9. Authentication Types

No Auth – Open APIs  
Basic – Authorization: Basic base64(username:pass)  
OAuth 2.0 – Google login  
Token – Authorization: Bearer ey123...



## 10. HTTP Headers

Content-Type – Body format (application/json)  
Accept – Response format  
Cookie – Session tracking  
Host – Domain (api.site.com)



## 11. Request Parameters

Header – Authorization  
Query – ?id=1&sort=asc  
Form – HTML form values  
Path – /user/{id} e.g., /user/10

In Postman, Pre-request scripts run before the request — useful for setting dynamic data like tokens.

Test scripts run after the response — used to validate status codes, content, or response time.

## 12. Sample JSON

{  
 "id": 1,  
 "name": "John"  
}

## 13. JSON Nested Arrays

{  
 "users": [  
 {  
 "id": 1,  
 "skills": ["Java", "API"]  
 }  
 ]  
}

## 14. Status Codes

100 – Continue  
200 – OK / Created  
300 – Redirect  
400 – Client Error (e.g., 404 Not Found)  
500 – Server Error (e.g., 500 Internal Server Error)

## 15. Tabs in Postman

Params, Authorization, Headers, Body, Scripts (pre-request, test), Settings

## 16. Query vs Path Params

Query: /user?id=10  
Path: /user/10

## 17. Why Newman

Run Postman collections via CLI, useful in CI/CD.

## 18. Postman Variables

Global – Available everywhere  
Collection – Within collection  
Environment – Based on selected env  
Local – Only in that request  
Data – Used in data-driven testing

## 19. Set Environment Variables

pm.environment.set("token", "abc123");

## 20. Run Scripts in Parallel

Use Newman with JSON iteration or external tools like npm parallel.

## 21. Generate Reports

newman run collection.json -r cli,html,json

## 22. Check Logs

Use console.log() and open Postman Console (Ctrl + Alt + C)

## 23. pm methods

pm.test("Status code is 200", function () {  
 pm.response.to.have.status(200);  
});  
  
pm.environment.set("token", "abc123");

## 24. Run Using Runner

Use Collection Runner > Select collection + environment + data file

## 25. Data Driven with CSV/JSON

Upload CSV/JSON to Runner, access with data.variable\_name

## 26. Can Postman Test Response Time & Headers?

pm.test("Response time < 200ms", function () {  
 pm.expect(pm.response.responseTime).to.be.below(200);  
});  
pm.test("Content-Type is JSON", function () {  
 pm.response.to.have.header("Content-Type");  
});

## 27. pm.environment.set vs pm.variables.set

environment.set – Updates environment variable  
variables.set – Updates local variable temporarily

## 28. Pre-request vs Test Script

Pre-request – Before call (e.g., token gen)  
Test – After response (e.g., validate status)

## 29. Schedule Collection Daily

Use Postman Monitor or Newman with CRON/Jenkins

## 30. Best Practices

• Use environments and variables  
• Validate responses  
• Assertions  
• Data-driven  
• Use version control  
• Monitor APIs  
• Automate CI/CD