# 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

a) All Web services are APIs but not all APIs are Web services.

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

c) 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.

d) 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.

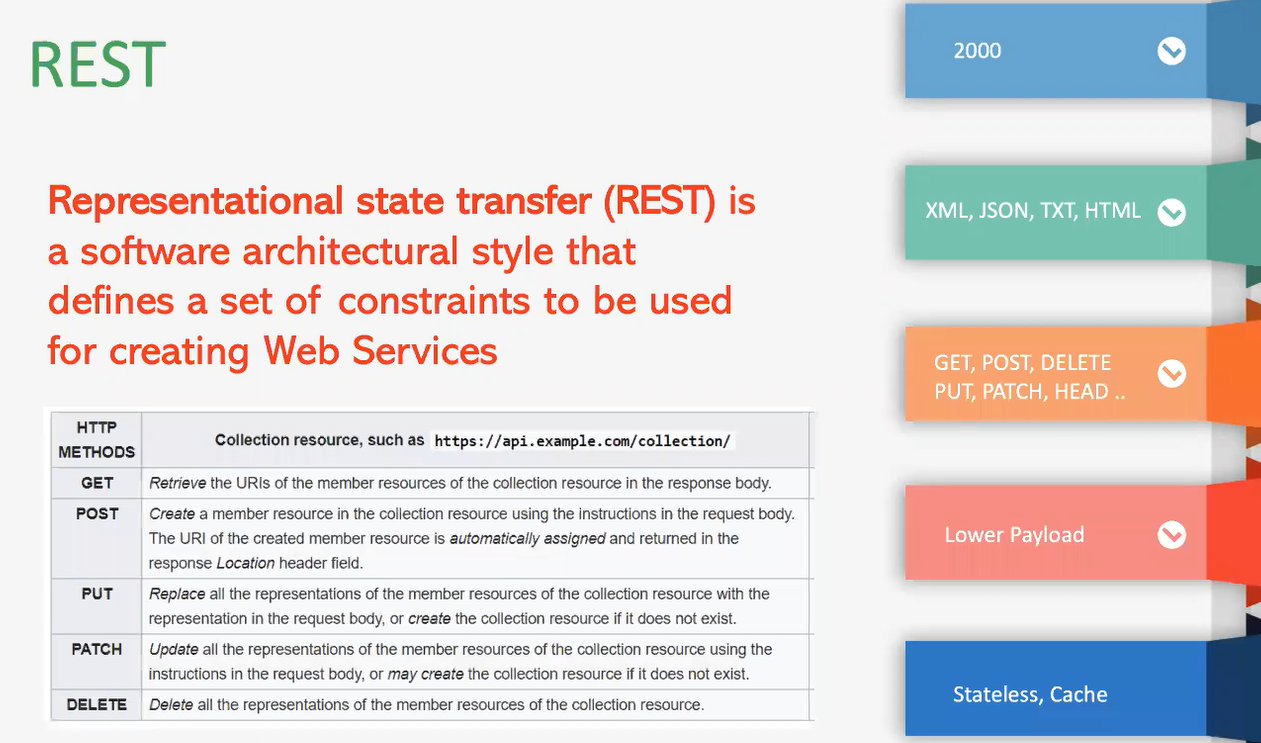
## 6. REST API 🡪 REST (Representational State Transfer) uses HTTP for operations. Example: GET /books/1 returns book details.

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

## 

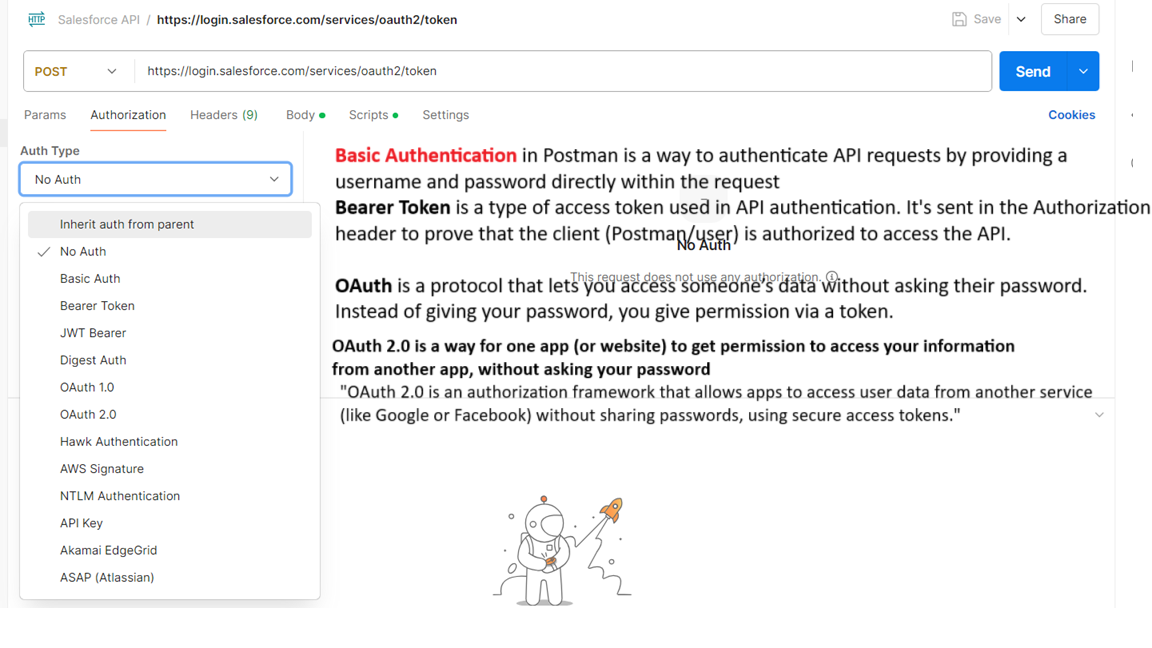
## 8. REST API

Lightweight, stateless, uses JSON, scalable.

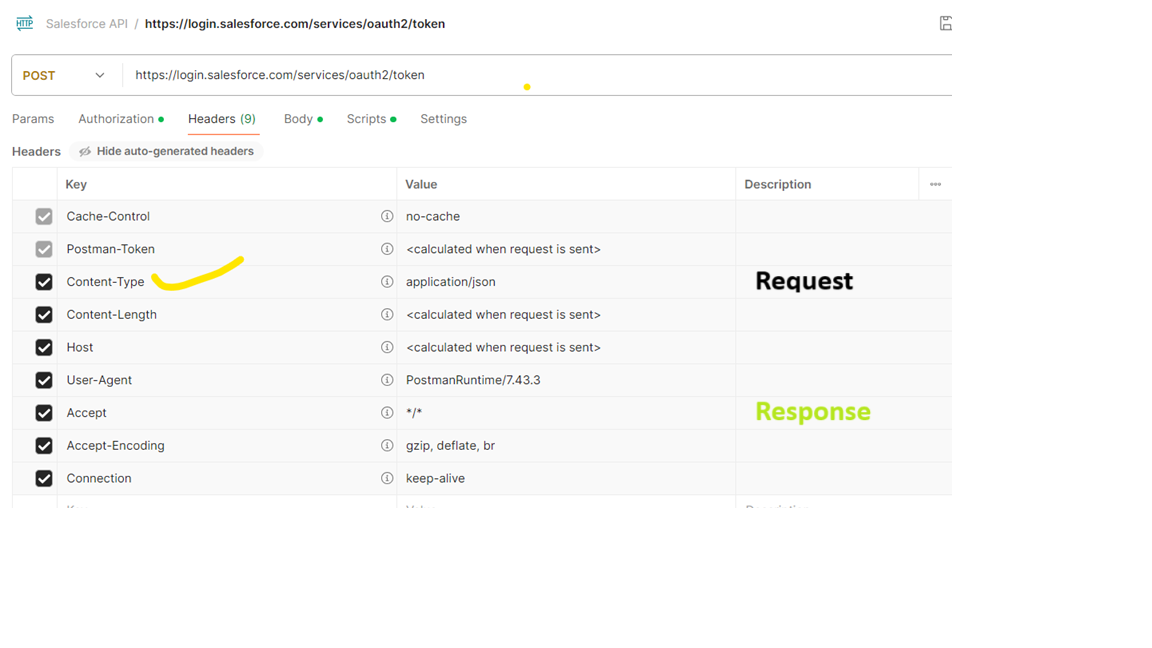
In the context of REST (Representational State Transfer), **statelessness** is a fundamental principle that dictates that each client-server interaction is independent and self-contained. This means that the server does not store any information about the client's previous requests. Each HTTP request from the client must contain all the information necessary for the server to understand and process the request.

**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.

Path params 🡪 Will append in URI  
Query 🡪 Will add in URI as Key and value pair

## 12. Sample JSON

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

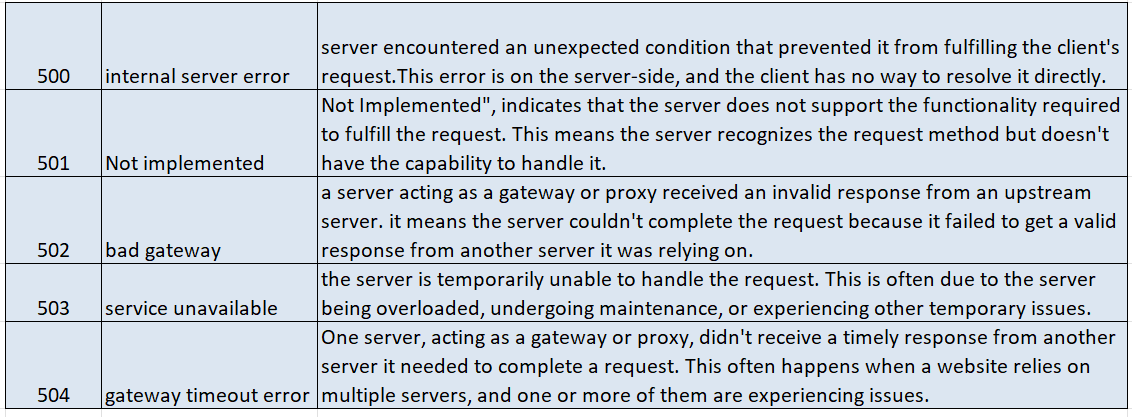
## 13. JSON Nested Arrays

{  
 "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

## Header ::

## 

**Scripts**  :: Pre request and post response .

A screenshot of a computer

AI-generated content may be incorrect.

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.

A screenshot of a computer code

AI-generated content may be incorrect.

# PostMan :: Postman is one of the most popular software testing tools which is used for API testing. With the help of this tool, developers can easily create, test, share, and document APIs. Introduction to Postman

* Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses.
* While using Postman, for testing purposes, one doesn't need to write any HTTP client network code. Instead, we build test suites called collections and let Postman interact with the API.
* In this tool, nearly any functionality that any developer may need is embedded. This tool has the ability to make various types of HTTP requests like GET, POST, PUT, PATCH, and convert the API to code for languages like JavaScript and Python.

## 16. Query vs Path Params

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

## 17. Why Newman

Run Postman collections via CLI, useful in CI/CD. command line based postman , especially if we want to integrate with Jenkins or any CI / CD Devopps tool.

## 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

**Global** – It is applicable for all different requests in postman (For multiple collections like – Service now collection, JIRA collection / Smoke collection / sanity etc).

**Collection** – Common variable for All request for specific collection

**Environment** – common for all requests for specific env – Test/ stage /dev, pre production

(When data is changing with respect to your environment)

**Data -** external file coming from csv or JSON

**Local –** locally created , with in Request

## 19. Set Environment Variables

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

Setting Global variable :: - **Settings To export data**

A screenshot of a computer

AI-generated content may be incorrect.

View icon – To set variable

Creating Global variable

A screenshot of a computer

AI-generated content may be incorrect.

Collection var –

Click on Edit for selected collection

A screenshot of a computer

AI-generated content may be incorrect.

Env variable and global var ::

A screenshot of a computer

AI-generated content may be incorrect.

Collection variable restricted to collection

A screenshot of a computer

AI-generated content may be incorrect.

***We can’t Run scripts/requests in parallelly in Postman, we can achieve this thru Newman and able to generate Allure Reports.* How to use Env variables in Request**

**Env var which we saved -** dev80088.service-now.com

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**After using all vars - GET**

**A screenshot of a computer

AI-generated content may be incorrect.**

**POST**

**A screenshot of a computer

AI-generated content may be incorrect.**

Delete

A screenshot of a computer

AI-generated content may be incorrect.

**PATCH ::**

**A screenshot of a computer

AI-generated content may be incorrect.**

## 20. Run Scripts in Parallel

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

**To pass sys\_id which is generated using POST, e need to pass it for some other requests to delete or update (PATCH) the same.**

**Pm – post man library , java script based library which is available in postman.**

**Test – function – tets and confirm whether postman is working fine or not .**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Writing test assertions**

The **pm** object provides most of the functionality for **testing** your request and response data.

You can use pm.test to write test specifications inside either the **Pre-request** or **Tests** scripts. Tests include a name and assertion—Postman will output test results as part of the response.

The pm.test method returns the pm object, making the call chainable. The following sample test checks that a response is valid to proceed.

pm.test("response should be okay to process", function () {

pm.response.to.not.be.error;

pm.response.to.have.jsonBody('');

pm.response.to.not.have.jsonBody('error');

})

JavaScript Object Notation (**JSON**) is a standard text-based format for representing structured data based on JavaScript object syntax. It is commonly used for transmitting data in web applications (e.g., sending some data from the server to the client, so it can be displayed on a web page, or vice versa

I have set of Requests like POST, GET and DELETE

For each **POST** request, one record will be generated and **sys\_id** will be generated for those record. And if we want to DELETE the record we can use the **sys\_id** as a parameter.

//confirm status of the code

Pm.test(“xxxxxxxx”,function()

{pm.response.to.have.status(200)}

)

//Get response (json) - (since we cant get in XML format we can use json format)

// Will use var or let to save response

var json= pm.response.jsosn();

var local\_sys\_id = json.result.sys\_id;

**// set id value as global var**

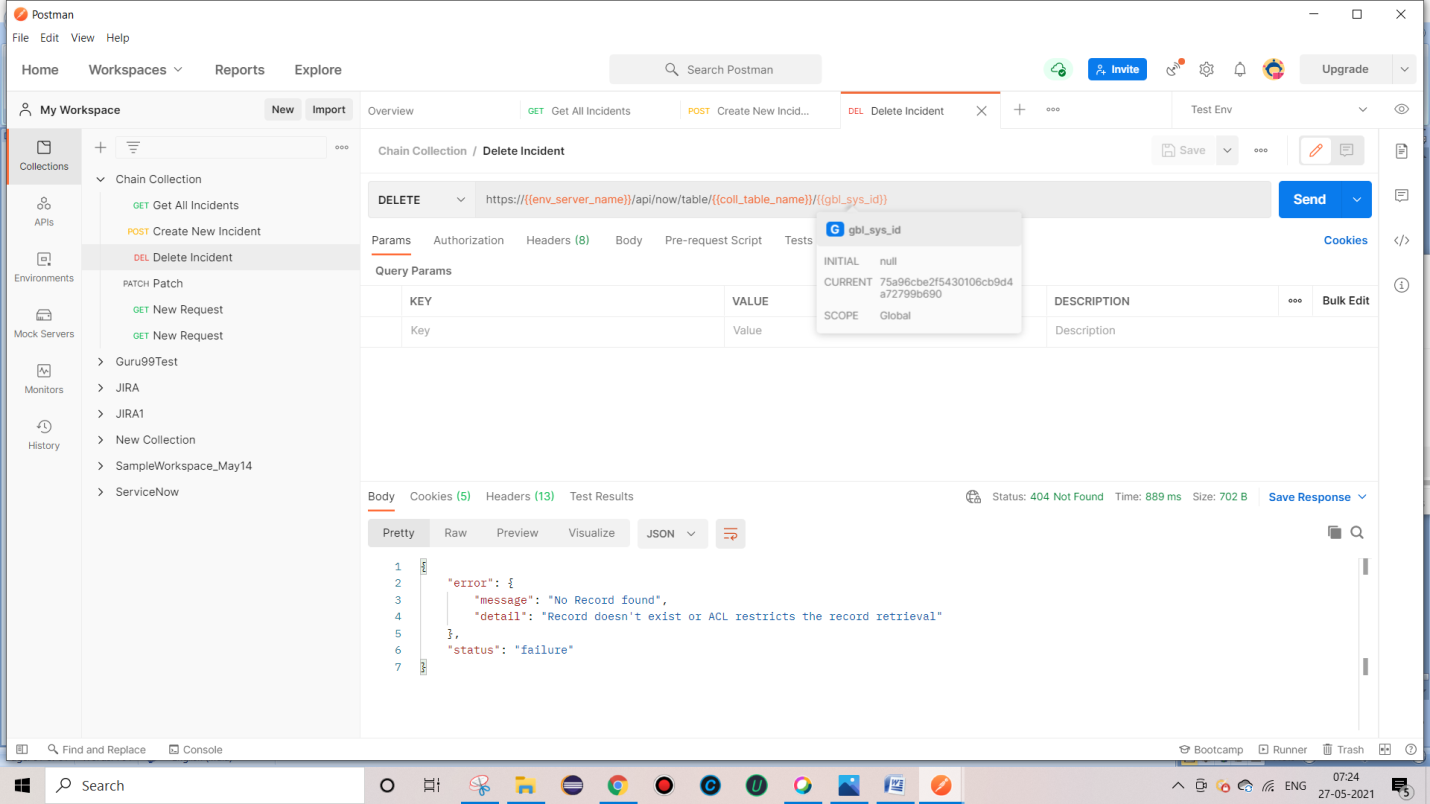
**Pm.globals.set(“()global\_sys\_id”,** local\_sys\_id**);**

**After creating record using POST Request**

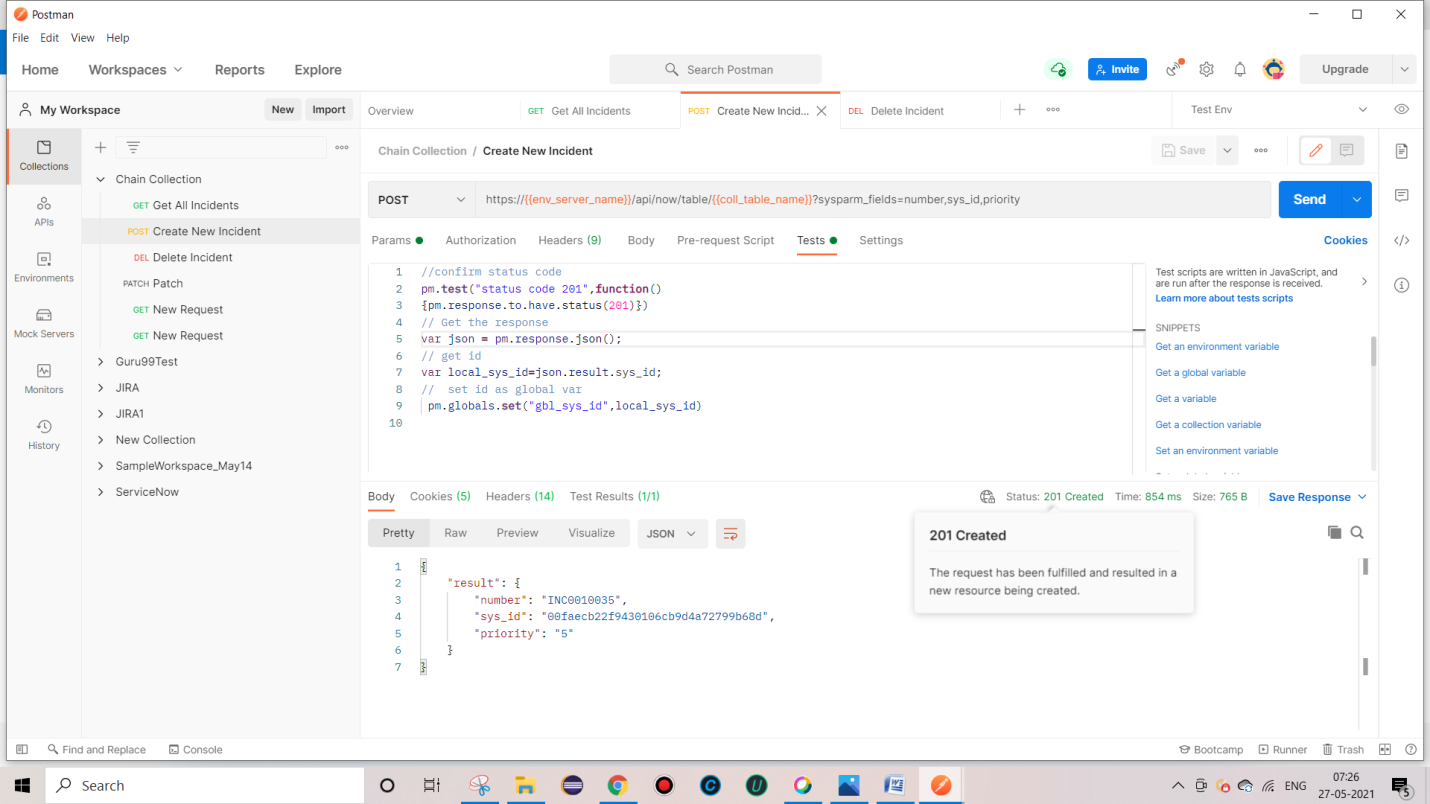
**A screenshot of a computer

AI-generated content may be incorrect.**

**Verify the sys\_id is passed into DELETE Request**

****

**POST – create :: 201 SUCCESS , created.**

****

**DELETE Record – 204 , Successful but not returning anything**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Try DELETE after deleting Record – 404 – Not Found since it is already deleted in previous request**

**A screenshot of a computer

AI-generated content may be incorrect.**

**In some scenarios, Get might get multiple records.**

**Here we got 94 records . A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.**

**Java script does not show any exception like java, it throw only undefined**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A green background with black text

AI-generated content may be incorrect.**

**To Run chain collections**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Sometimes we may need to work with diff data , To pass that data , We can use .csv or .json file**

**So that we can create multiple records with multiple data passed from .csv or .json file.**

1. **Go to Collection and click on “Run”**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Add this under “ Body ” tab**

**A screenshot of a computer

AI-generated content may be incorrect.**

**3. Select the Request which we want o Run multiple times using csv file and since my csv file having 5 set of data ,Iterations showing as “5” .**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Csv File**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Click on ”Run Chain” collection**

**A screenshot of a computer

AI-generated content may be incorrect.**

**We even see the csv/ json file data by click on Preview button**

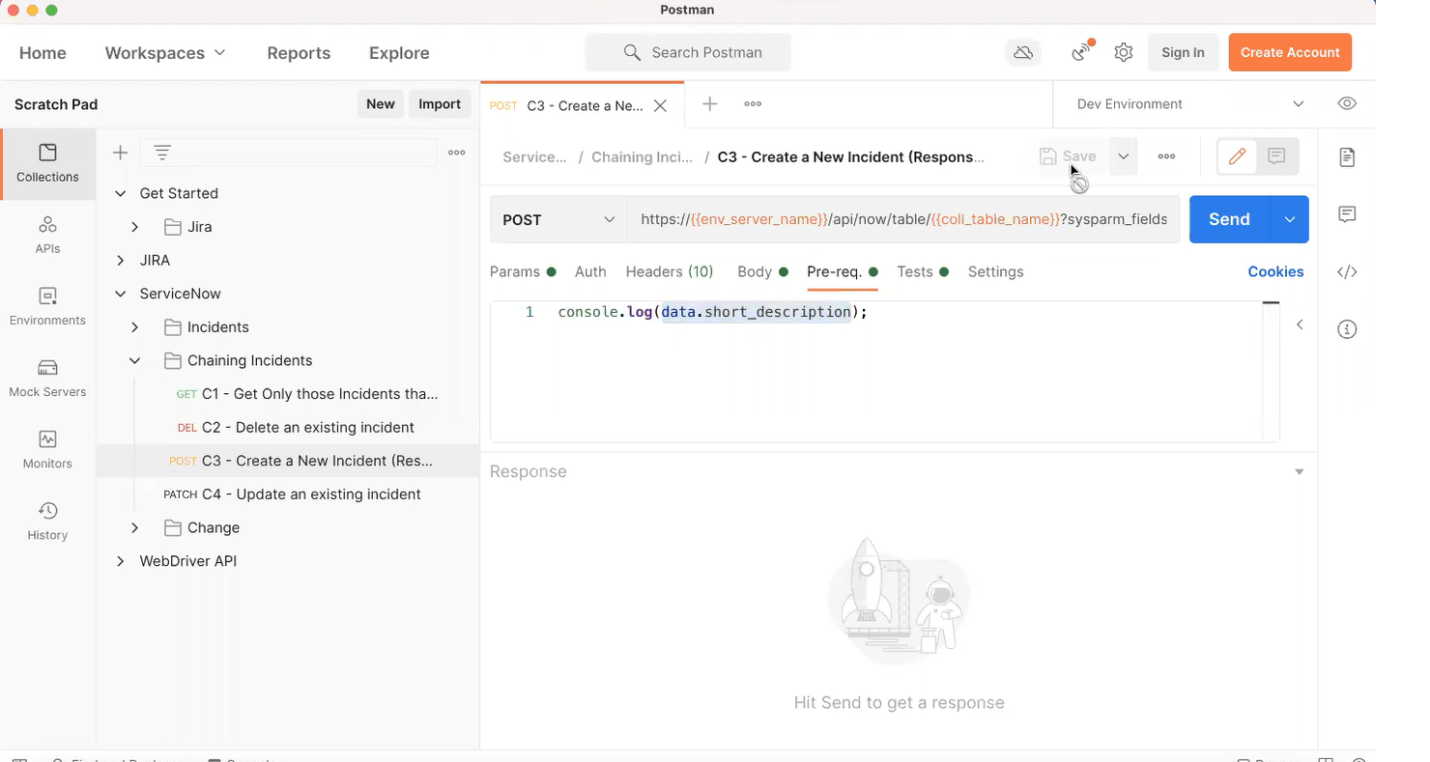
**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**And if we want to print data, we can add like this**

**1 Runner – only at collection level**

**View icon 🡪Edit**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Export**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Working with Newman – Newman built on node**

1. **Make sure All ur files in .json format**

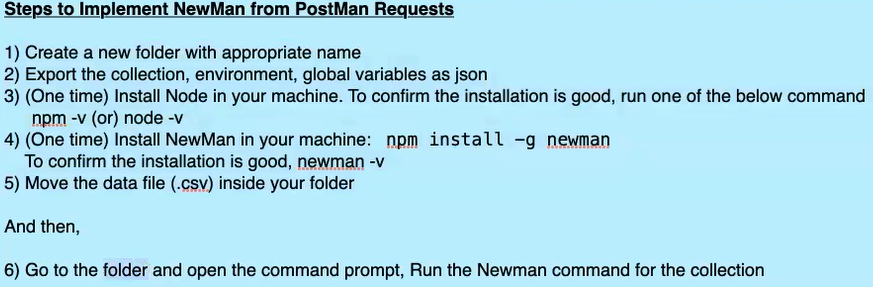
**Export**

1. **Collection**
2. **Env variable**
3. **Global variable**

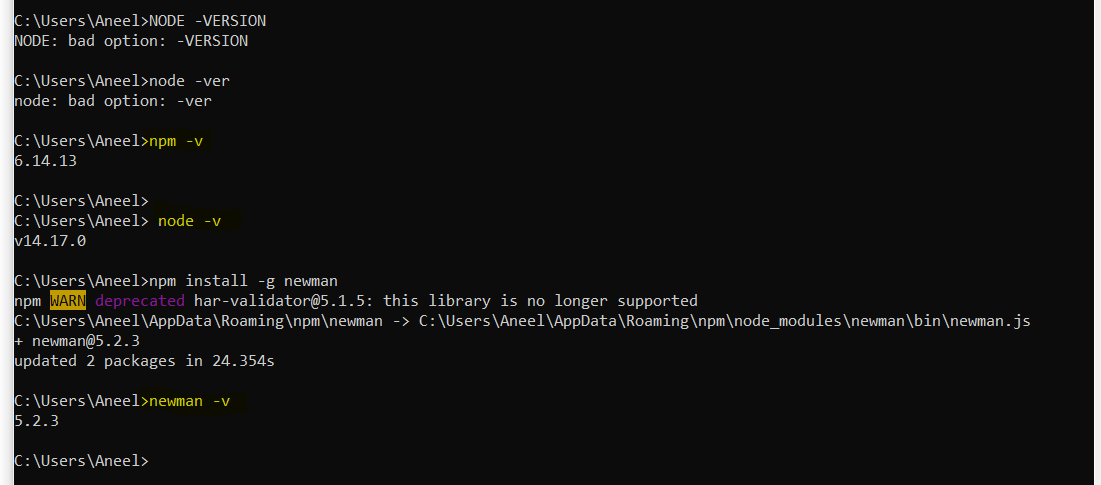
**A screenshot of a computer

AI-generated content may be incorrect.**

**Steps to work with Newman**

****

**Verify node and**

****

**Click on cmd from the Folder**

**A screenshot of a computer

AI-generated content may be incorrect.To run New ManA screenshot of a computer

AI-generated content may be incorrect.** **C:\Users\Aneel\Desktop\API\NewMann>newman**

**run ChainCollection.postman\_collection.json -g global.json -e env.json -d POSTData.csv**

**( Collection folder global var env var DataFile )**

## A screenshot of a computer screen AI-generated content may be incorrect. 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

## 31. Working with Jenkins