# Postman API Guide

PPT link: <https://www.popai.pro/share.html?shareKey=48ff431f73ecb39ae921ec7c65218e841c2216eb10b0c8980463e4d7952dbc04&utm_source=presentationsharepage&inviteCode=IN_7noqhoA6KjQ>

## What is API

API (Application Programming Interface) is a set of protocols, tools, and definitions that allow different software applications to communicate with each other. It acts as an intermediary, enabling seamless interaction between various systems, services, or components.

\*\*Example:\*\* A weather application fetching real-time weather data from a remote weather API.

## Types of API

1. \*\*Open APIs (Public APIs):\*\* Available for public use without restrictions.

- Example: OpenWeatherMap API.

2. \*\*Partner APIs:\*\* Shared with specific business partners, requiring authentication.

- Example: PayPal Partner API.

3. \*\*Internal APIs (Private APIs):\*\* Used within organizations for internal processes.

- Example: Internal employee management API.

4. \*\*Composite APIs:\*\* Combine multiple APIs into a single API to streamline processes.

- Example: An API that handles user registration and payment simultaneously.

---

## Why We Mostly Use REST API Rather Than SOAP

1. \*\*Simplicity:\*\* REST uses standard HTTP methods (GET, POST, PUT, DELETE), making it easier to implement.

2. \*\*Scalability:\*\* REST APIs are stateless, which helps in scaling the application.

3. \*\*Flexibility:\*\* REST supports multiple formats like JSON, XML, and HTML, while SOAP only uses XML.

4. \*\*Performance:\*\* REST APIs are lightweight and faster compared to SOAP.

## Types of API Requests

1. \*\*GET:\*\* Retrieve data from the server.

- Example: Fetching user details.

2. \*\*POST:\*\* Send data to the server to create a resource.

- Example: Creating a new user.

3. \*\*PUT:\*\* Update an existing resource.

- Example: Modifying user details.

4. \*\*DELETE:\*\* Remove a resource from the server.

- Example: Deleting a user account.

5. \*\*PATCH:\*\* Partially update a resource.

- Example: Updating a user’s email.

## HTTP Response Codes

1. \*\*2xx Success:\*\* Indicates successful requests.

- Example: 200 OK, 201 Created.

2. \*\*3xx Redirection:\*\* Indicates the client must perform additional actions.

- Example: 301 Moved Permanently.

3. \*\*4xx Client Errors:\*\* Indicates an error on the client side.

- Example: 400 Bad Request, 404 Not Found.

4. \*\*5xx Server Errors:\*\* Indicates issues on the server side.

- Example: 500 Internal Server Error.

**Definition of Authorization in Postman**

Authorization in Postman refers to the process of verifying the identity of a user or client before allowing access to a resource, API, or system. It ensures that only authenticated and authorized users can interact with an API.

Types of Authorization in Postman

Postman supports multiple types of authorization methods:

No Auth – No authentication is required.

API Key – A unique key passed in the request header or query parameter.

Bearer Token – A token used to access protected resources.

Basic Auth – Uses a username and password encoded in Base64.

Digest Auth – A more secure version of Basic Auth, using encryption.

OAuth 1.0 & 2.0 – A widely used authorization method for web applications.

Hawk Authentication – A cryptographic authentication scheme.

AWS Signature – Used for authentication with AWS APIs.

NTLM Authentication – Used in Microsoft environments.

## Benefits of POSTMAN and Why We Use It

1. \*\*User-Friendly Interface:\*\* Simplifies API testing.

2. \*\*Automation:\*\* Supports creating test scripts and running them automatically.

3. \*\*Collaboration:\*\* Enables team collaboration by sharing collections.

4. \*\*Environment Variables:\*\* Simplifies testing by storing configurations.

5. \*\*Pre-Request Scripts:\*\* Allows execution of scripts before sending requests.

6. \*\*Mock Servers:\*\* Create mock APIs for testing purposes.

7. \*\*Testing Automation:\*\* Integration with CI/CD tools like Jenkins.

8. \*\*Performance Testing:\*\* Perform load testing to check API efficiency.

**Here’s a step-by-step guide on how to install Postman with official download links:**

**Step 1: Download Postman**

Visit the official Postman website:

https://www.postman.com/downloads/

Click on Download for your operating system (Windows, macOS, or Linux).

**Step 2: Install Postman (Windows)**

After downloading the .exe file, double-click it to start installation.

Follow the on-screen instructions.

Once installed, launch Postman.

**Step 3: Install Postman (Mac)**

Open the .dmg file you downloaded.

Drag and drop the Postman app into the Applications folder.

Open Postman from Launchpad or Finder.

**Step 4: Install Postman (Linux)**

**Download the tar.gz file.**

**Extract the files using:**

tar -xvzf Postman-linux-x64.tar.gz

**Move it to the /opt directory:**

sudo mv Postman /opt/

**Create a symlink for easy access:**

sudo ln -s /opt/Postman/Postman /usr/bin/postman

**Launch Postman by typing postman in the terminal.**

Step 5: Sign Up and Start Using Postman

Open Postman and sign up or log in with your account.

Start creating API requests and testing APIs!

# Postman Pre-request & Post-response Scripts Guide

## 1. Pre-request Scripts

Pre-request scripts in Postman are scripts that run before the request is sent. They are commonly used for:

* Setting up authentication (generating tokens)
* Setting variables dynamically
* Logging data before request execution

### Example Usage

// Set a dynamic timestamp

pm.environment.set("timestamp", new Date().toISOString());

// Generate a random user ID

pm.environment.set("user\_id", Math.floor(Math.random() \* 1000));

// Log details

console.log("Pre-request script executed");

### Common Use Cases

| **Use Case** | **Example** |
| --- | --- |
| **Generate Authentication Token** | Use pm.sendRequest() to fetch an auth token and set it for use in the request. |
| **Set Dynamic Variables** | pm.environment.set("varName", "value") |
| **Logging Data** | console.log("Executing pre-request script") |

## 2. Post-response (Tests) Scripts

Post-response scripts, also known as tests, run after a request is executed. They help with:

* Validating response data
* Storing values from the response
* Running automated assertions

### Example Usage

// Check if the response status is 200

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

// Validate response time is less than 500ms

pm.test("Response time is under 500ms", function () {

pm.expect(pm.response.responseTime).to.be.below(500);

});

// Extract and store response data as environment variables

let jsonData = pm.response.json();

pm.environment.set("user\_token", jsonData.token);

### Common Use Cases

| **Use Case** | **Example** |
| --- | --- |
| **Validate Status Code** | pm.response.to.have.status(200); |
| **Check Response Time** | pm.expect(pm.response.responseTime).to.be.below(500); |
| **Extract Response Data** | pm.environment.set("token", pm.response.json().token); |
| **Check JSON Keys** | pm.expect(jsonData).to.have.property("id"); |

## What is Environment in POSTMAN

An environment in Postman is a set of variables that can be used to customize requests. Environments make it easy to switch between different configurations, like development, staging, and production.

\*\*Example:\*\* Defining variables like `{{baseUrl}}` and `{{apiKey}}`.

---

## What is Monitor(Scheduling) in POSTMAN and How We Can Perform It

A Postman Monitor is a tool that runs API collections periodically to check their performance, uptime, and correctness.

\*\*Steps to Perform Monitoring:\*\*

1. Create a collection of API requests.

2. Go to the "Monitors" section in Postman.

3. Configure the schedule, environment, and region.

4. Start the monitor to get reports on performance and errors.

## What is Load API Testing in POSTMAN and How We Can Perform It

Load testing evaluates the performance of an API under heavy loads.

\*\*Steps to Perform Load Testing in Postman:\*\*

1. Create a collection of APIs.

2. Use Postman’s "Collection Runner" to send multiple requests in parallel.

3. Monitor response times and errors.

4. Use external tools like Newman or Postman CLI for large-scale load testing.

## What is Pipeline in POSTMAN and How We Can Perform It

A pipeline in Postman refers to automating API workflows by chaining multiple requests.

\*\*Steps to Create a Pipeline:\*\*

1. Use variables to pass data between requests.

2. Set pre-request and test scripts to handle logic.

3. Execute the pipeline using the "Collection Runner" or Newman.

## Additional Benefits of POSTMAN

1. \*\*Built-In Code Generator:\*\* Generate client code in multiple languages.

2. \*\*Version Control:\*\* Track changes to collections.

3. \*\*Custom Workspaces:\*\* Organize API collections effectively.

4. \*\*Integration with Tools:\*\* Seamless integration with Jenkins, GitHub, and more.

5. \*\*Secure API Testing:\*\* Offers encryption and secure storage for sensitive data.

# Demo API Collection

## Overview

This document provides a comprehensive analysis of the API endpoints in the "Real Working API Collection." The collection is structured in accordance with Postman Collection Schema version 2.1.0. Below is a detailed breakdown of each API endpoint, including requests, responses, and associated tests.

**How to import curl and import json in postman**

To import a cURL request into Postman:

1. Copy the cURL command from your terminal or API documentation.
2. Open Postman, click on "Import" (top left), and select "Raw Text".
3. Paste the cURL command, click "Continue", then "Import" to convert it.

To import an API into Postman:

1. Click "Import" and select "File", "Folder", "Link", or "Raw text".
2. Upload the Postman Collection JSON file or enter the API documentation link.
3. Click "Import" to load the API requests into your workspace.

## API Endpoints

### Get User

Description: Retrieves user details based on the user ID.

Method: GET

Here baseUrl= https://jsonplaceholder.typicode.com

URL: {{baseUrl}}/users/{{userId}}

Tests:

* - Status code is 200.
* - Response includes the user ID matching the request.

### Create Post

Description: Creates a new post for a specified user.

Method: POST

URL: {{baseUrl}}/posts

Headers:

* - Content-Type: application/json

Request Body:

{"title": "foo", "body": "bar", "userId": {{userId}}}

Tests:

* - Status code is 201.
* - Response contains a `post ID`.
* - The `post ID` is stored for future requests.

### Update Post

Description: Updates an existing post.

Method: PUT

URL: {{baseUrl}}/posts/{{postId}}

Headers:

* - Content-Type: application/json

Request Body:

{"id": {{postId}}, "title": "updated title", "body": "updated body", "userId": {{userId}}}

Tests:

* - Status code is 200.
* - Title is successfully updated.

### Delete Post

Description: Deletes a specified post.

Method: DELETE

URL: {{baseUrl}}/posts/{{postId}}

Tests:

* - Status code is 200.

### Authenticate User

Description: Authenticates a user and retrieves a token.

Method: POST

URL: https://reqres.in/api/login

Headers:

* - Content-Type: application/json

Request Body:

{"email": "eve.holt@reqres.in", "password": "cityslicka"}

Tests:

* - Status code is 200.
* - Response contains a `token`.
* - The token is stored for future requests.

### Get Secured Resource

Description: Fetches user data for a secured resource.

Method: GET

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

Headers:

* - Authorization: Bearer {{authToken}}
* - Content-Type: application/json

Tests:

* - Status code is 200.
* - Response contains user data.

## Running the Collection with Newman

Newman is a powerful command-line collection runner for Postman. It allows you to automate API testing and generate detailed reports in various formats. Here's how to use it:

Before Installing Newman We need to install Node Here is the installation Commands:

1. Download the installer from node official site
2. Open command prompt Then check the Version for node by run the following command
3. node -v
4. npm -v

### Installation

To use Newman, ensure that Node.js and npm are installed on your system. Then, install Newman globally using the following command:

npm install -g newman-reporter-htmlextra

newman run DemoApi.json -r htmlextra

Running the Collection

Run the collection by executing the command below. This command is ready to paste directly into your terminal:

newman run "Real Working API Collection.postman\_collection.json" -r cli,html --reporter-html-export "newman-report.html"

### Explanation of the Command

- `newman run`: Executes the specified collection.

- `Real Working API Collection.postman\_collection.json`: The name of the collection file to be executed.

- `-r cli,html`: Specifies that both CLI and HTML reports should be generated.

- `--reporter-html-export "newman-report.html"`: Saves the HTML report to a file named `newman-report.html` in the current directory.

### Benefits of Using Newman

- Automation: Automate API tests as part of CI/CD pipelines.

- Detailed Reports: Generate detailed CLI and visual reports for better analysis.

- Customizability: Combine Newman with custom scripts for advanced workflows.

## Summary

The "Real Working API Collection" offers a robust set of endpoints covering CRUD operations for posts, user authentication, and secured resource access. The included tests ensure that responses meet the expected structure and status codes, enabling efficient debugging and validation during API development.