

## Postman for API Testing: A Comprehensive Guide for QA Testers

Asha Rani Rajendran Nair Chandrika\*

**Citation:** Chandrika ARR. Postman for API Testing: A Comprehensive Guide for QA Testers. *J Artif Intell Mach Learn & Data Sci* 2022, 1(1), 1514-1518. DOI: doi.org/10.51219/JAIMLD/asha-ranirajendran/340

**Received:** 03 March, 2022; **Accepted:** 28 March, 2022; **Published:** 30 March, 2022

\***Corresponding author:** Asha Rani Rajendran Nair Chandrika, Infosys Ltd, India

**Copyright:** © 2022 Chandrika ARR. Postman for API Testing: A Comprehensive Guide for QA Testers., This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

### ABSTRACT

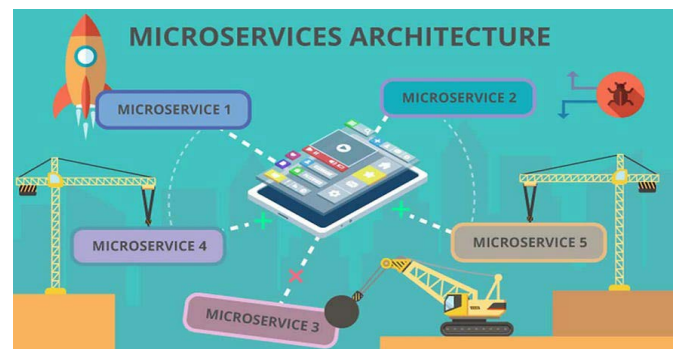
API testing has become a critical element in modern software development, especially with the rise of interconnected systems and microservices. Postman, a popular tool for testing APIs, offers an intuitive interface to help QA testers manually or automatically validate API responses, workflows and performance. This article explores the fundamentals of API testing using Postman, focusing on essential features such as collections, environment variables, test automation with JavaScript scripting and integration into CI/CD pipelines. By following the steps outlined in this guide, testers can streamline their API testing process, enhance test reliability and ensure consistent API functionality. The article also discusses best practices in API testing, including test organization, automation and validation techniques, offering practical insights that can be applied immediately in real-world scenarios.

### 1. Introduction

In today's fast-evolving technological landscape, service-oriented backend architectures, such as microservices, have become the backbone of modern enterprises. These architectures rely on distributed services that communicate via **Application Programming Interfaces (APIs)**, making robust and reliable APIs essential to business success. As businesses scale, APIs must evolve to meet dynamic and complex requirements and ensuring their quality is paramount. Efficient **API testing** helps maintain fast development cycles without compromising code quality, allowing teams to release updates confidently while ensuring seamless communication between services.

Postman, a widely used platform for API testing and development, is instrumental in facilitating both manual and automated API testing. As microservices architecture becomes more common, the need to validate APIs effectively has grown significantly. Postman offers an intuitive interface for sending requests, validating responses and automating testing workflows, making it a critical tool for QA testers. This guide will explore how

Postman's powerful features can be leveraged to ensure that APIs are reliable, efficient and bug-free, supporting comprehensive testing for both isolated services and integrated systems.



**Figure 1:** Microservice Architecture<sup>1</sup>.

### 2. What is Postman?

Postman is a versatile and user-friendly tool that allows developers and testers to interact with APIs by sending requests and receiving responses. Postman supports various HTTP

methods, including GET, POST, PUT, DELETE, PATCH and many others, making it ideal for testing RESTful APIs.

Postman allows users to organize their API tests into collections, manage environment variables and automate tests using scripting. Its intuitive interface makes it a go-to choose for both manual and automated API testing.

### 2.1. Installing and Setting up Postman

#### Step 1: Download Postman

Postman is available for download as a desktop application for Windows, macOS and Linux. Simply visit Postman’s official website and download the version that matches your operating system.

#### Step 2: Installation

After downloading, follow the installation instructions for your OS. Once installed, you can either create a Postman account to sync your work across devices or use it without signing in.

#### Step 3: Getting Started

Upon launching Postman, you’ll be greeted with a clean and organized interface. The left panel is where you can create collections and folders to manage your API requests. The main panel is where you’ll compose and send requests to your APIs.

### 2.2. Understanding API Testing

API testing is a form of software testing that focuses on verifying the functionality, performance and security of APIs. Unlike user interface testing, which focuses on the frontend of an application, API testing deals with the backend, ensuring that the communication between software components works as expected.

Key aspects of API testing include:

- **Functionality Testing:** Verifying that the API functions as intended.
- **Performance Testing:** Measuring response times, request handling under load, etc.
- **Security Testing:** Ensuring that sensitive data is protected and unauthorized access is prevented.
- **Reliability Testing:** Ensuring consistent API behavior across different requests and inputs.

### 2.3. Making API Requests with Postman

Creating a new request in Postman is a fundamental step for interacting with APIs and essential for efficient API testing and debugging. Postman’s intuitive interface allows you to make various types of HTTP requests like GET, POST, PUT and DELETE, enabling real-time communication with APIs. In this guide, we will walk you through the process of creating and organizing new requests into collections, setting up the request URL, selecting the appropriate HTTP method and analyzing the server’s response. Additionally, you’ll learn how to save and reuse these requests, streamlining the API testing workflow and improving collaboration with your team.

#### Steps for Making a Request:

- Click on “New”:
- In the header section of Postman, click the “NEW” button.
- Choose “Request”:

- From the available options, click on “Request.”
- Name Your Request:
- Enter a meaningful name for the request (e.g., First API).

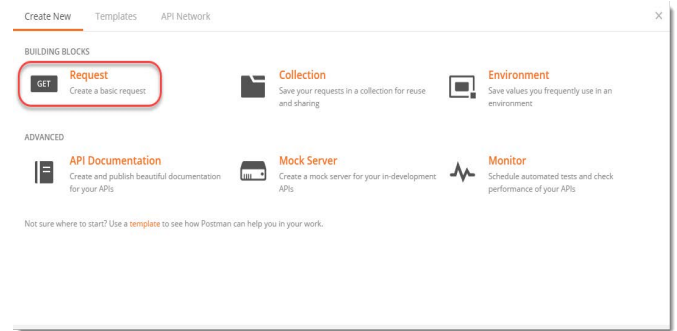


Figure 2: Creating a Request.

1. **Create or Select a Collection:**
2. In the bottom panel, either select an existing collection or create a new one by entering a name (e.g., First Collection).
3. Click “+Create Collection” to save the request under this collection.

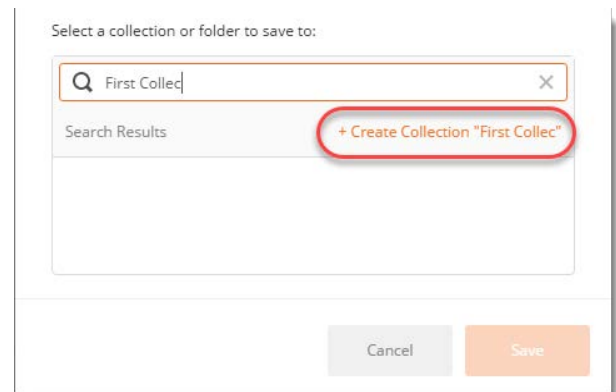


Figure 3: Creating the Collection

4. Access the Request:
5. Go to the “Collections” tab in the sidebar, select the collection you just created (e.g., First Collection) and then select the request (e.g., First API) within that collection.

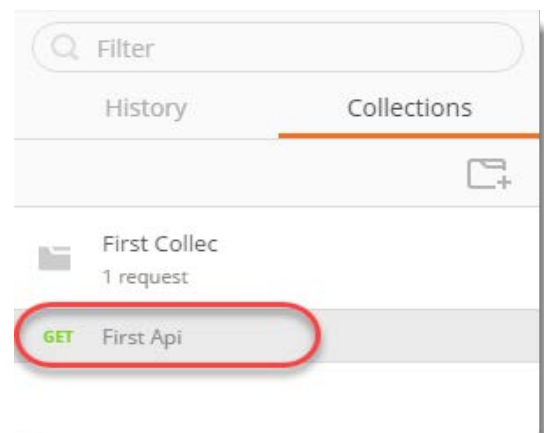


Figure 4: Selecting the collection for the request.

6. Enter the Request URL:
7. In the address bar, enter the URL (e.g., [www.google.com](http://www.google.com)) for the API you are testing.

8. Send the Request:
9. Press the “Send” button to make the request.
10. Save the Request:
11. You can either overwrite the existing request by pressing “Save” or use the “Save As” option to create a new version of the request.



Figure 5: Sending the request.

12. View the Response:
13. The response from the server will appear in the response section. Pay attention to the status code (e.g., 200 OK for a successful request) and the response body.

**2.4. Key Elements of a Response in Postman**

Postman offers a user-friendly interface to inspect responses, displaying important data like the **status code**, **response time**, **size** and the **body content**. Without requiring much effort, testers can extract meaningful insights from the response to validate the accuracy of the server’s actions.

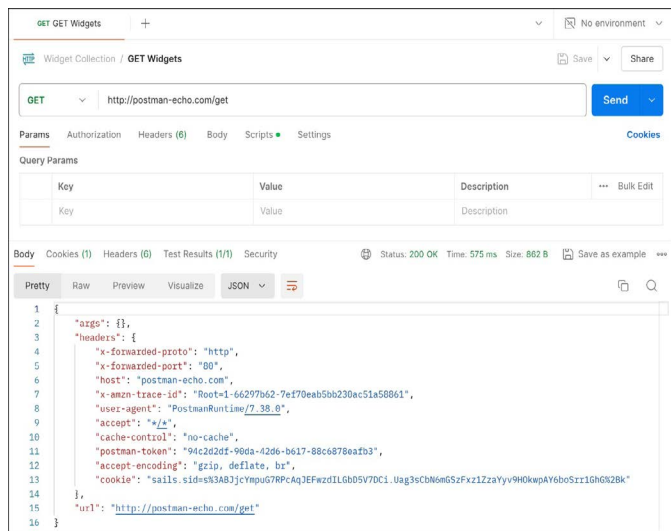


Figure 6: Response in Postman.

**Response Status and Information:**

- **Status Code:** Each response includes a status code, indicating the result of the request (e.g., 200 for a successful request, 400 for a bad request). These codes help determine if the request was processed correctly or if there were errors.
- **Time:** This refers to how long it took for the server to process the request and send a response. This duration is important in meeting **Service Level Agreements (SLAs)** for web services.
- **Size:** Postman also shows the approximate size of the response, including headers and body content, which helps understand the load on the server and the data sent back.

**Response Body:**

The body contains the actual content of the response from the server. Postman allows viewing the body in various formats:

- **Pretty:** A color-coded, well-structured display for easy readability.
- **Raw:** Displays the raw data received from the server without any additional formatting.
- **Preview:** Provides a view of how the response would look if rendered in a browser, helpful for checking webpage responses.

**Response Headers:**

Headers provide additional information about the response. For instance, the **Content-Type** header indicates the format of the response (e.g., JSON or HTML), while the **Server** header shows details about the server processing the request.

**Response Cookies:**

Cookies store information sent from the server, often to track user sessions or personalize experiences. Postman makes it easy to view these cookies for inspection during testing.

**2.5. API Testing using Postman and Newman**

Newman is a command-line interface (CLI) tool designed for running Postman collections directly from the terminal or command prompt. According to Postman’s official site, Newman is described as a **command-line Collection Runner** for Postman. It enables developers to execute Postman collections outside the Postman app, making it especially useful for automation and integration with Continuous Integration (CI) workflows. Newman can run collections in the same way they are executed inside Postman, but from the command line, allowing for easier scripting and integration.

**2.6. Running a Collection with Newman:**

**Install Newman:**

Since Newman is built on Node.js, it uses the npm (Node Package Manager) for installation.

For installing Newman in your system, follow these steps.

1. Open the command prompt (\*Terminal for mac)
2. Type `npm install -g newman`.
3. This will install a new dependency through NPM. You will see the following screen after pressing enter
4. (if npm is successfully fetched and installed).

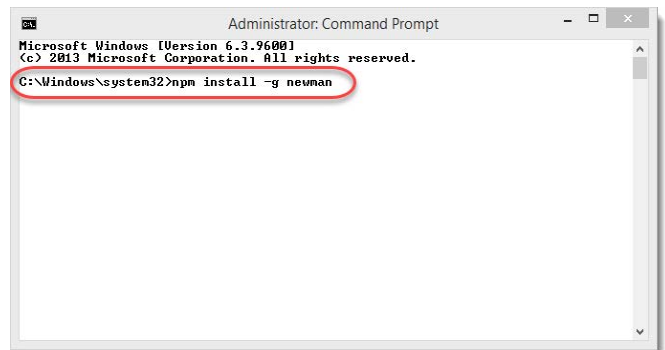


Figure 7: Installation of Newman.

**2.7. Running the collection using Newman through JSON file**

1. Click on the export link alongside the collection name and click on **export** in the following panel. Export feature is used to export the collection by saving it to your computer

and which can be later share with the team over the Email in a zip file.

2. Save the Json file in your system and remember the directory.
3. Once you save the Json file, visit the shell of your system and change the current directory to the directory in which you saved this Json file.
4. After changing the directory, run the following command -`newman run <name of the file>`
5. Press enter and you will see the expected results of your collection Newman collection.

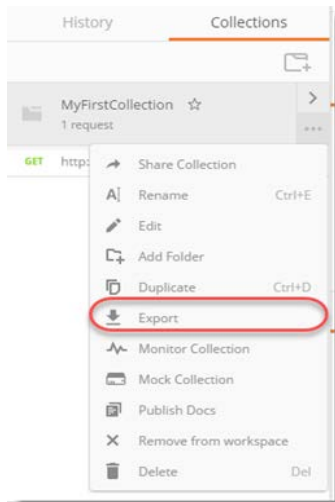


Figure 8: Export Collection Name.

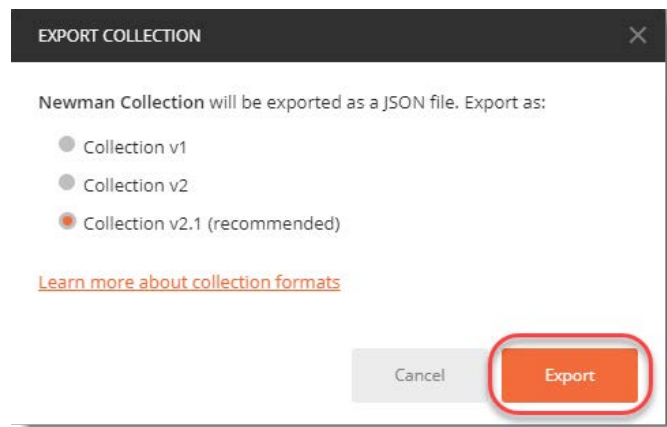


Figure 9: Export Newman Collection.

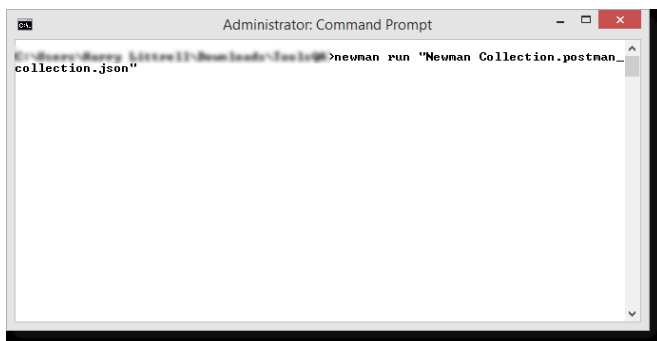


Figure 10: Installation of Newman.

### 2.8. Integrating Postman with CI/CD Pipelines

Postman collections can be integrated with CI/CD tools to ensure API tests are run automatically whenever code is deployed. Tools like Jenkins or GitLab CI can trigger Newman

to execute tests after each deployment.

By integrating API tests into your CI/CD pipeline, you ensure that any changes to your API or backend services are tested automatically, reducing the risk of introducing bugs into production.

### 2.9. Run Postman Collection on Jenkins

- To run a Collection, go to your **Postman Collection** and get the **Share Link**
- Go to the **Jenkins Job** and under **build section**,
- write the command to run the Collection: `newman run "<link>"`



Figure10: Run Postman collection.

Save the changes and click on **build now**. This will start the Jenkins job, which will perform all the actions defined with in the job. As we have specified an action to run the *Postman Collection using Newman command*, this will do the same.

Check the **Console Output** in the Jenkins.

### 3. Conclusion

- **User-Friendly Interface:** Postman provides a straightforward and intuitive interface for API testing, catering to both beginners and experienced testers.
- **Robust Testing Features:** It offers a wide range of features, including request creation, response validation and test script execution, which enhance the efficiency of API testing.
- **Automation Capabilities:** Postman’s integration with Newman allows for automated testing of API collections, facilitating continuous integration and continuous delivery (CI/CD) practices.
- **Collaboration and Sharing:** The platform enables teams to collaborate effectively by sharing collections, environments and results, promoting better communication between developers and testers.
- **Support for Multiple Protocols:** Postman supports various protocols such as REST, SOAP and GraphQL, making it versatile for testing different types of APIs.

### 4. References

1. <https://medium.com/assertqualityassurance/rest-api-test-automation-with-postman-jenkins-1-of-3-860edf3c2a45>
2. API Testing and Development with Postman: A practical guide to creating, testing and managing APIs for automated software testing -by Dave Westerveld
3. API Testing with Postman by KC Martin
4. <https://blog.postman.com/continuous-api-testing-with-postman/>
5. <https://learning.postman.com/docs/getting-started/first-steps/sending-the-first-request/>
6. <https://toolsqa.com/postman/response-in-postman/>
7. <https://dzone.com/articles/rest-api-tutorials>

8. <https://blog.postman.com/how-to-write-powerful-automated-api-tests-with-postman-newman-and-jenkins/>