Introduction

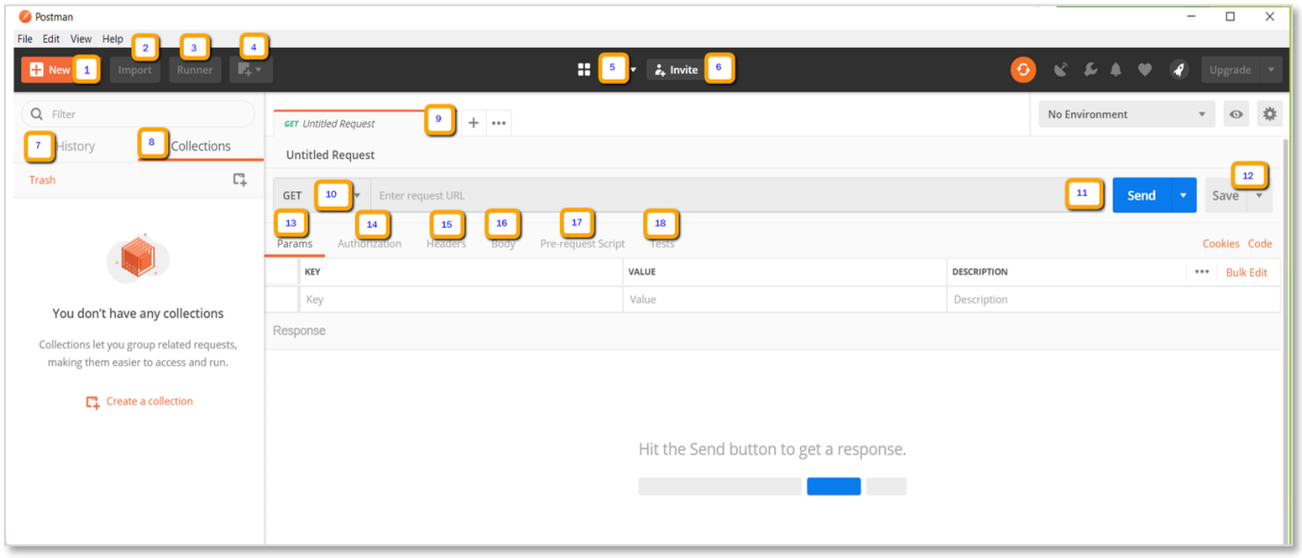
Postman is a tool to test the RESTful APIs created by others or by yourself. This tool offers a sleek user interface to make HTML requests, without the hassle of writing the code just to test an API's functionality.

TCPWave uses the Postman tool over other RESTful API testing tools because of the following:

- Accessibility - To use Postman, you are required to log-in to the respective accounts making it easy to access files anytime, anywhere if a Postman application is installed on the computer.
- Use of Collections - Postman lets users create collections for their API calls. Each collection can create subfolders and multiple requests. This helps in organizing your test suites.
- Collaboration - Collections and environments can be imported or exported making it easy to share files. A direct link can also be used to share collections.
- Creating Environments - Having multiple environments aids in less repetition of tests as one can use the same collection but for a different environment
- Creation of Tests - Test checkpoints such as verifying for successful HTTP response status can be added to each API call which helps ensure test coverage.
- Automation Testing - Using the Collection Runner or Newman, tests can be run in multiple iterations saving time for repetitive tests.
- Debugging - Postman console helps to check what data has been retrieved making it easy to debug tests.
- Continuous Integration - With its ability to support continuous integration, development practices are maintained.

Using Postman

The below image describes the various options available in the Postman UI to seamlessly use Postman tool:



1. New - This is where you can create a new request, collection, or environment.

2. Import - This is used to import a collection or environment. There are options such as import from file, folder, link, or paste the raw text.

3. Runner - Automation tests can be executed through the Collection Runner.

4. Open New - Open a new tab, Postman Window or Runner Window by clicking this button.

5. My Workspace - You can create a new workspace individually or as a team.

6. Invite - Invite users to your team to collaborate with them

7. History - Past requests are displayed in History. This makes it easy to track the actions that you have done.

8. Collections - Organize your test suite by creating collections. Each collection may have subfolders and multiple requests. You can also duplicate a request or a folder.

9. Request tab - This displays the title of the request you are working on. By default, the "Untitled Request" would be displayed for requests without titles.

10. HTTP Request - Clicking this would display a dropdown list of different requests such as GET, POST, COPY, DELETE, etc. In testing, the most used requests are GET and POST.

11. Request URL - Also known as an endpoint. You can identify the link to where the API will communicate with.

12. Save - If there are changes to a request, you are required to click save so that new changes are not lost or overwritten.

13. Params - This is where you will write parameters needed for a request such as key values.

14. Authorization - To access APIs, proper authorization is needed. It may be in the form of a username and password, bearer token, etc.

15. Headers - You can set headers such as content type JSON depending on the needs of the organization.

16. Body - This is where one can customize details in a request commonly used in POST requests.

17. Pre-request Script - These are scripts that will be executed before the request. Usually, pre-request scripts for the setting environment are used to ensure that tests will be run in the correct environment.

18. Tests - These are scripts executed during the request. It is important to have tested as it sets up checkpoints to verify if response status is ok, retrieved data is as expected and other tests.

Integrate Postman with TCPWave

1. Import JSON environment file. You can declare global variables in that file.

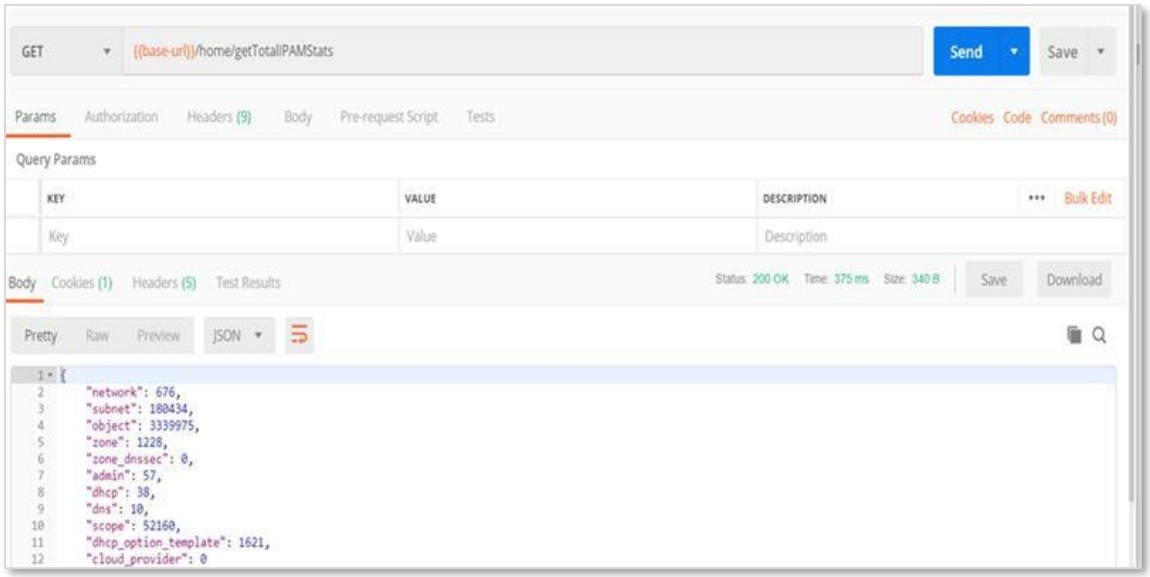
2. The imported global variables are as shown:

3. TCPWave uses Token-Based Authentication to execute the Postman calls.

4. For token-based authentication, you need to create the session token in the TCPWave IPAM and copy-paste the value in the environment file token field value.

5.For another Global variable base-URL you are required to provide the address of the system. The address of the system belongs to the system on which execution is done. (URL: https://www.tcpwave.com:7891/tims/rest)

Example: Get Call



Example: Post Call: