A Comprehensive API Testing Guide

What is an API?

API is an acronym and it stands for Application Programming Interface. API is a set of routines, protocols, and tools for building Software Applications. APIs specify how one software program should interact with other software programs.

Routine: a program that performs a particular task. Routine is also known as procedure, function, or subroutine.

Protocols: A format for transmitting data between two systems.

In simple words, API acts as an interface between two software applications and allows the two software applications to communicate with each other.

Let's see some examples of an API in a more approachable way.

Assume an **API** as a **Waiter** at a **Restaurant**.

At a restaurant, you give an order based on the items available on the menu, and the waiter tells the kitchen. The kitchen prepares your food, and the waiter brings it back to you.

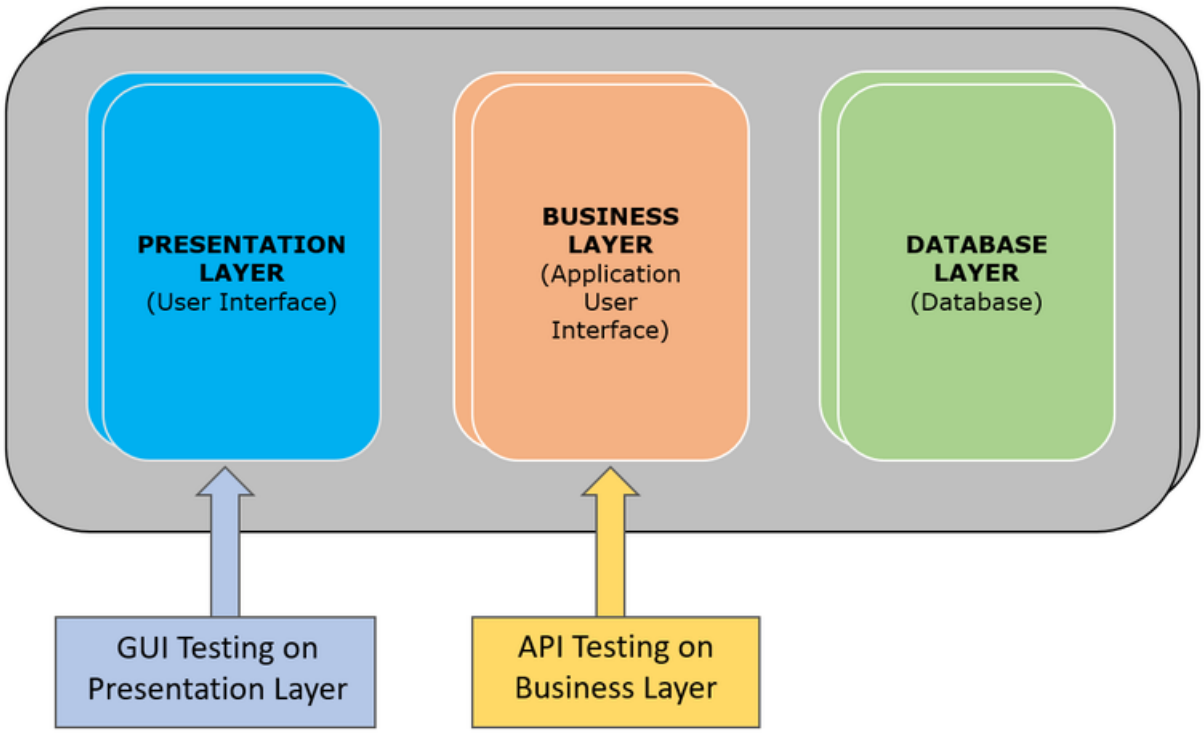
In this scenario, the waiter's role is similar to an API. As a waiter, the API takes a request from a source, takes that request to the database, fetches the requested data from the database, and returns a response to the source.

Now let's see another example.

If you are using a flight service engine say Expedia, where you search for flights on a specific date. Once you pass the data such as Source, Destination, Onward Date, and Return Date and click on search. Expedia sends a request to airlines through an API as per your search details. The API then takes the airline's response to your request and delivers it right back to Expedia.

API gets the request from the user and gives the response without exposing internal logic. API acts like an Abstraction in the OOPs concept.

What is API Testing?



API testing is a type of software testing that involves testing APIs directly and also as a part of integration testing to check whether the API meets expectations in terms of functionality, reliability, performance, and security of an application. In API Testing our main focus will be on a Business logic layer of the software architecture.

API testing can be performed on any software system which contains multiple APIs. API testing won't concentrate on the look and feel of the application. API testing is entirely different from GUI Testing.

Let's see how is UI testing is not similar to API testing?

UI (User Interface) testing is to test the graphical interface part of the application. Its main focus is to test the look and feel of an application. On the other hand, API testing enables communication between two different software systems. Its main focus is on the business layer of the application.

Why is API Testing Important?

API testing is crucial for several reasons.

- First, it ensures that the core functionality of the application works as intended by validating the interactions between different software systems. This early detection of issues can save significant time and resources during the development process.
- Second, because APIs often handle large amounts of data and execute critical tasks, thorough testing helps maintain data integrity and security, protecting against potential breaches or data losses.
- Third, well-tested APIs contribute to a seamless user experience by ensuring that backend processes run smoothly and quickly, ultimately leading to more robust and reliable software applications. Finally,
- API testing supports continuous integration and delivery practices, enabling faster and more efficient release cycles by automating tests and identifying defects early on.

API Testing Types

API testing typically involves the following practices:

Functional Testing: This type assesses whether the API performs and functions as expected. It ensures the API returns the correct responses for a given request, maintaining the integrity and capability of the API.

Load Testing: Load testing examines how the API handles a large number of requests and data. It evaluates the API's performance under heavy loads to ensure it remains reliable during peak usage times.

Security Testing: This testing checks the API for vulnerabilities and ensures that it is secure. It helps in identifying potential threats and weaknesses that could be exploited by malicious users.

Validation Testing: Validation testing ensures that the API meets the specified requirements and performs actions correctly. It validates that the API works as intended with the correct input and output.

Error Handling Testing: This type of testing involves checking how the API manages errors and exceptions. It ensures the API provides meaningful error messages and handles exceptions appropriately without crashing.

Performance Testing: Performance testing measures the speed, responsiveness, and stability of the API. It helps to identify any performance bottlenecks and ensures that the API works efficiently under various conditions.

Reliability Testing: This ensures that the API consistently performs well over time. It checks if the API remains stable and reliable even after long periods of use.

Unit Testing: This tests individual parts of the API in isolation. It ensures that each component works as expected on its own.

Integration Testing: Integration testing examines how the API interacts with other APIs, systems, or services. It ensures that the combined parts of the application work together as expected.

Regression Testing: Regression testing is conducted to ensure that recent code changes have not adversely affected existing functionalities of the API. It helps maintain the overall stability after updates or improvements.

Fuzz Testing: This involves sending random or unexpected data to the API to see how it handles it. It helps find vulnerabilities and improves the robustness of the API.

Fault Tolerance Testing: This tests how well the API handles unexpected problems or errors. It checks if the API can continue working even when issues occur.

Interoperability and WS Compliance testing: Interoperability and WS Compliance Testing is a type of testing that applies to SOAP APIs. Interoperability between SOAP APIs is checked by ensuring conformance to the Web Services Interoperability profiles. WS-* compliance is tested to ensure standards such as WS-Addressing, WS-Discovery, WS-Federation, WS-Policy, WS-Security, and WS-Trust are properly implemented and utilized

API Documentation Testing: This checks that the documentation for using the API is accurate and easy to understand. Good documentation helps developers use the API correctly.

Common tests on APIs

Some of the common tests we perform on APIs are as follows.

- To verify whether the return value is based on the input condition. The response of the APIs should be verified based on the request.
- To verify whether the system is authenticating the outcome when the API is updating any data structure
- To verify whether the API triggers some other event or request another API
- To verify the behavior of the API when there is no return value

Advantages of API Testing

- API Testing is **time effective** when compared to GUI Testing. API test automation requires **less code** so it can provide faster and better test coverage.
- API Testing helps us to **reduce** the testing cost. With API Testing we can find minor bugs before the GUI Testing. These minor bugs will become bigger during GUI Testing. So finding those bugs in the API Testing will be cost-effective to the Company.
- API Testing is language independent.
- API Testing is quite helpful in testing Core Functionality. We can test the APIs without a user interface. In GUI Testing, we need to wait until the application is available to test the core functionalities.
- API Testing helps us to reduce the risks.

What exactly needs to be verified in API Testing?

Basically, on API Testing, we send a request to the API with the known data and we analyze the response.

- Data accuracy
- HTTP status codes
- Missing or duplicate functionality
- Response time
- Reliability issues
- Error codes in case API return any errors
- Authorization checks

- Multithreaded issues
- Error codes if API returns
- Non-functional testing such as performance testing, security testing

What are API Methods?

API methods, often known as HTTP methods, are the actions that can be performed on the data provided by an API. The most commonly used API methods are GET, POST, PUT, and DELETE.

- **GET**: This method is used to retrieve data from a server. It does not change any data and is only used to fetch information.
- **POST**: This method is used to send data to a server to create a new resource. It usually results in adding new data to the server.
- **PUT**: This method is used to update an existing resource on the server. It sends data to the server to overwrite the existing information.
- **DELETE**: This method is used to remove an existing resource from the server. It tells the server to delete the specified data.

These methods help in performing various operations on the data, making it easier to manage and manipulate through APIs.

Difference between API testing and Unit Testing?

UNIT TESTING:

- Unit testing is conducted by the Development Team
- Unit testing is a form of White box testing
- Unit testing is conducted prior to the process of including the code in the build
- Source code is involved in Unit testing
- In unit testing, the scope of testing is limited, so only basic functionalities are considered for testing

API TESTING:

- API testing is conducted by QA Team
- API testing is a form of Black box testing
- API testing is conducted after the build is ready for testing

- Source code is not involved in API testing
- In API testing, the scope of testing is wide, so all the issues that are functional are considered for testing

Challenges in API testing

Some of the challenges we face while doing API testing are as follows

- Selecting proper parameters and its combinations
- Categorizing the parameters properly
- Proper call sequencing is required as this may lead to inadequate coverage in testing
- Verifying and validating the output
- Due to the absence of GUI, it is quite difficult to provide input values

Types of bugs we face when performing API testing:

Issues observed when performing API testing are

- Stress, performance, and security issues
- Duplicate or missing functionality
- Reliability issues
- Improper messaging
- Incompatible error handling mechanism
- Multi-threaded issues
- Improper errors

API Testing Best Practices

- Test for the expected results
- Add stress to the system by sending a series of API load tests
- Group API test cases by test category
- Create test cases with all possible inputs combinations for complete test coverage
- Prioritize API function calls to make it easy to test
- Create tests to handle unforeseen problems
- Automate API testing wherever it is possible

Postman Tutorial for Beginners: API Testing using Postman

What is API?

API is an acronym and it stands for **A**pplication **P**rogramming **I**nterface. API is a set of routines, protocols, and tools for building Software Applications. APIs specify how one software program should interact with other software program.

In simple words, API stands for **A**pplication **P**rogramming **I**nterface. API acts as an interface between two software applications and allows the two software applications to communicate with each other. API is a collection of software functions which can be executed by another software program.

What is Postman?

Postman is a **collaboration platform** for API development. It is a popular API client and it enables you to design, build, share, test, and document APIs.

Using the Postman tool, we can send HTTP/s requests to a service, as well as get their responses. By doing this we can make sure that the service is up and running.

Being originally a Chrome browser plugin, Postman now extends their solution with the native version for both Mac and Windows.

Why Postman?

Postman has become a tool of choice for over 8 million users.

- **Free:** It is free to download and use for teams of any size.
- **Easy:** Just download it and send your first request in minutes.
- **APIs Support:** You can make any kind of API call (REST, SOAP, or plain HTTP) and easily inspect even the largest responses.
- **Extensible:** You can customize it for your needs with the Postman API.

- **Integration:** You can easily integrate test suites into your preferred CI/CD service with Newman (command line collection runner)
- **Community & Support:** It has a huge community forum

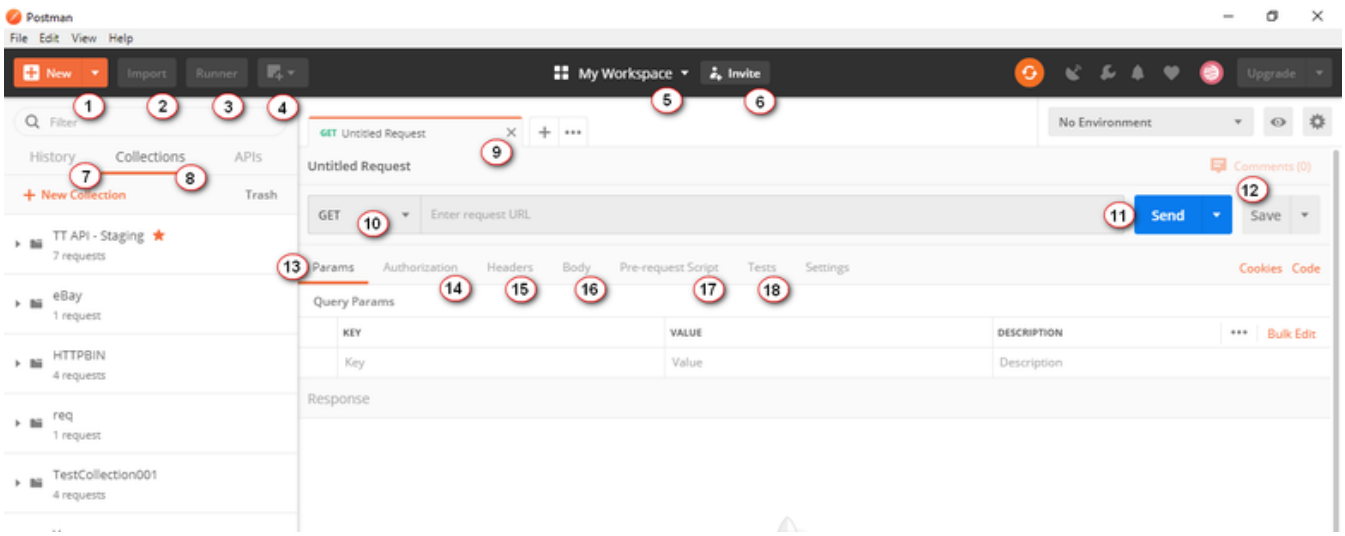
Postman Features:

Some of the features Postman tool offers are as follows.

- Easy-to-use REST client
- Rich interface which makes it easy to use
- Can be used for both manual and automated API testing
- Can be run on Mac, Windows, Linux & Chrome Apps
- Has a bunch of integrations like support for Swagger & RAML formats
- Has Run, Test, Document and Monitoring Features
- Doesn't require learning a new language
- Enable users to easily share the knowledge with the team as they can package up all the requests and expected responses, then send to their colleagues.
- Can be integrated with CI-CD tools like Jenkins, TeamCity etc.,
- Comes with a detailed API documentation
- API development & Automating API tests execution

How To Use Postman

All the components in the Postman tool have its own importance. Let's see the different options of Postman workspace now.



1. New: It is to create a new request, collection, or environment.

2. Import: It is to import a collection or environment. You can also find other options such as import from file, folder, link, or paste raw test.

3. Runner: We can execute automation tests using Collection Runner.

4. Open New: We can open Postman Window or Runner Window
5. My Workspace: It is your workspace. You can create a new workspace using this. A workspace is a shared context for building and consuming APIs. It allows real-time collaboration within and between teams.
6. Invite: It is to invite your team members to collaborate on a workspace.
7. History: Automatic saving of requests and responses in history which helps you track your past actions easily.
8. Collections: It is to organize and keep track of related requests.
9. Request tab: Title of the request you are working on. By default it is named as 'Untitled Request'.
10. HTTP Request: You can see requests like GET, POST, COPY, DELETE, etc.
11. Request URL: Here we mention the link to where the API will communicate with. It is also known as endpoint.
12. Save: To save the request or to update the existing request.
13. Params: We mention the parameters (key values) needed for a request.
14. Authorization: APIs use authorization to ensure that client requests access data securely. We mention authorization details like username, password, bearer token etc., here.
15. Headers: Some APIs require you to send particular headers such as JSON, JavaScript etc., along with requests, typically to provide additional metadata about the operation you are performing. You can set these up here
16. Body: It allows you to specify the data you need to send with a request. You can send various different types of body data to suit your API.
17. Pre-request Script: Pre-request scripts are written in JavaScript, and are run before the request is sent. This is perfect for use-cases like including the timestamp in the request headers or sending a random alphanumeric string in the URL parameters.
18. Tests: Tests are the scripts executed during the request. Tests allow you to ensure that your API is working as expected, to establish that integrations between services are functioning reliably, and to verify that new developments haven't broken any existing functionality.

Building Blocks of Postman:

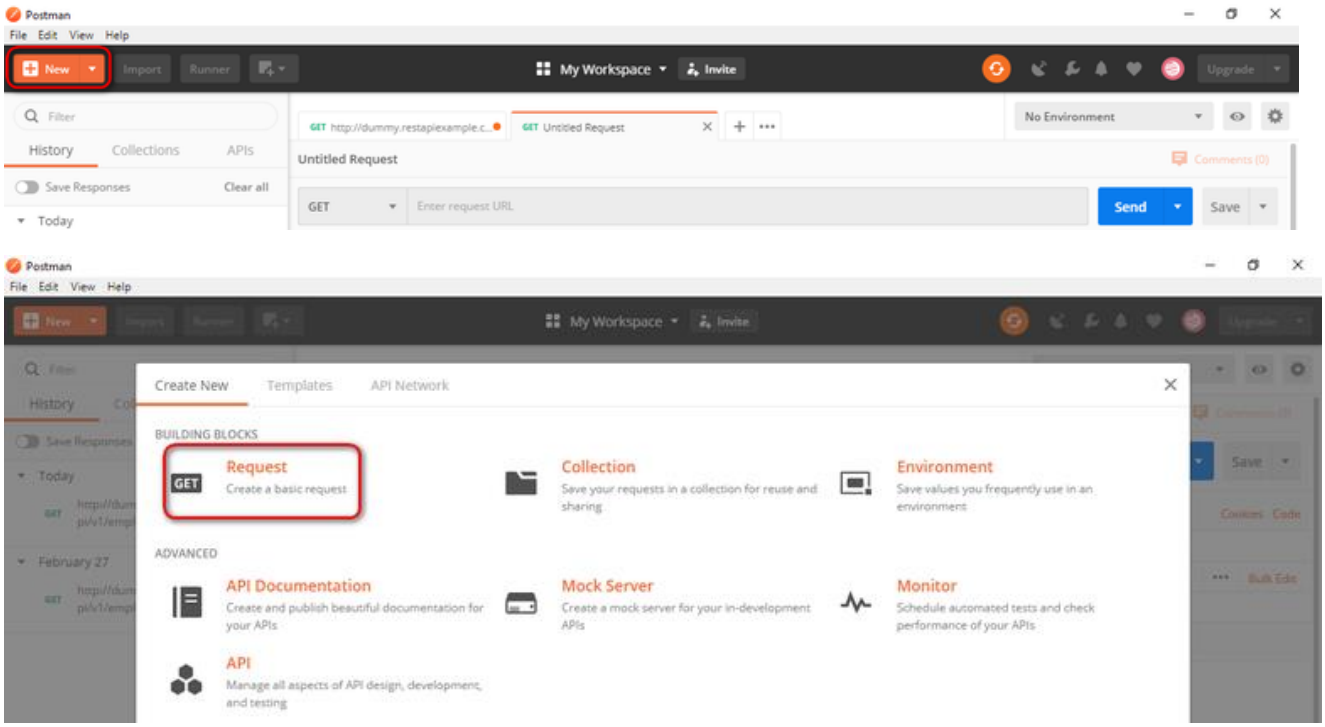
Before testing an API, first we will see some building blocks of Postman Tool that are essential for every Postman operations.

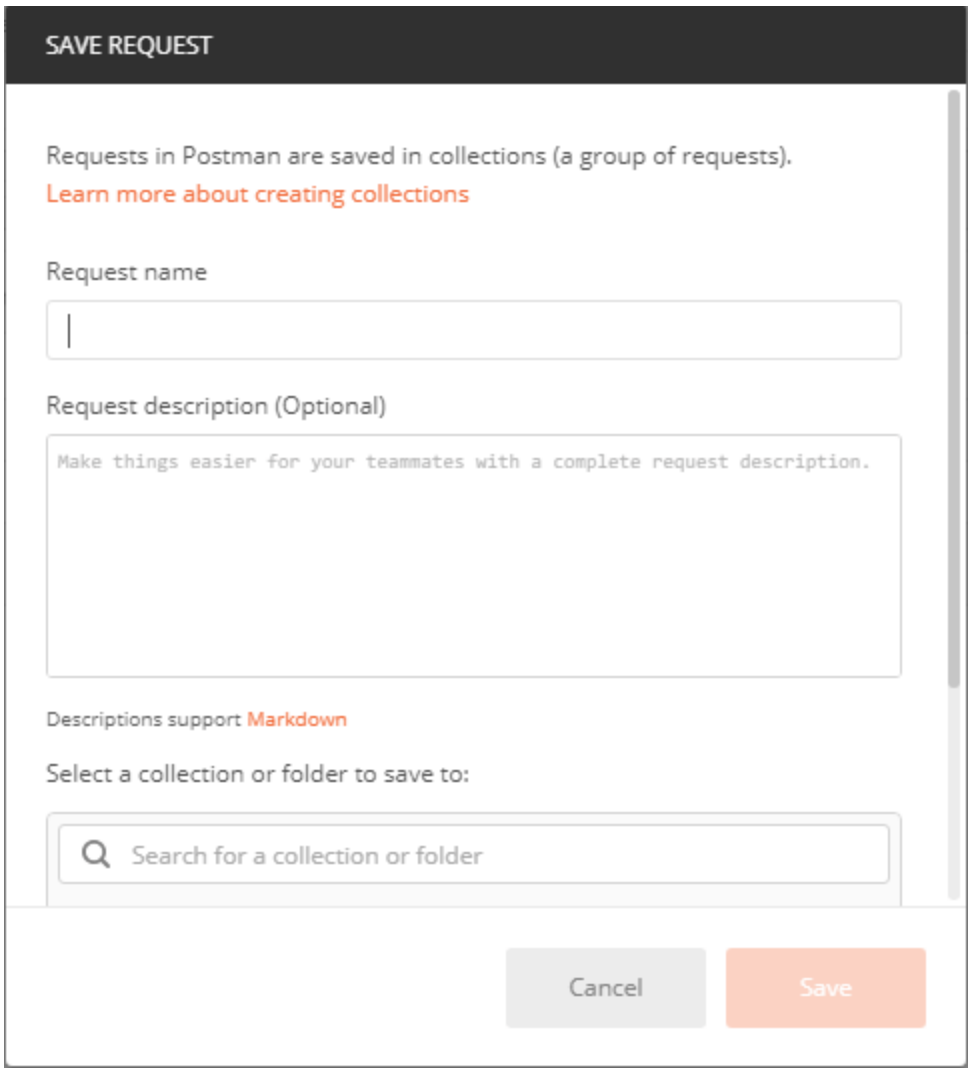
1. Requests
2. Collections
3. Environment

1. Requests:

A request is a combination of the URL, HTTP headers, Body or Payload. In the postman tool, you can save your requests and use them in the future based on your needs.

Click on ***New – Request***





You can make requests to APIs in Postman. An API request allows you to retrieve data from a data source, or to send data. APIs run on web servers, and expose endpoints to support the operations client applications use to provide their functionality.

Each API request uses an HTTP method.

What is HTTP?

HTTP stands for Hyper Text Transfer Protocol. HTTP enables communication between clients and servers. Clients are often web browsers and Servers are often computers on the cloud.

If a client submits an HTTP request to the server, then the server returns a response to the client. The response sent by the server contains status information about the request and the requested content.

Most commonly used HTTP methods are as follows:

1. GET: GET method is used to retrieve data from an API.
2. POST: POST method is used to send new data to an API
3. PUT: PUT method is used to update existing data
4. PATCH: PATCH method is used to update existing data
5. DELETE : DELETE method is used to remove existing data.

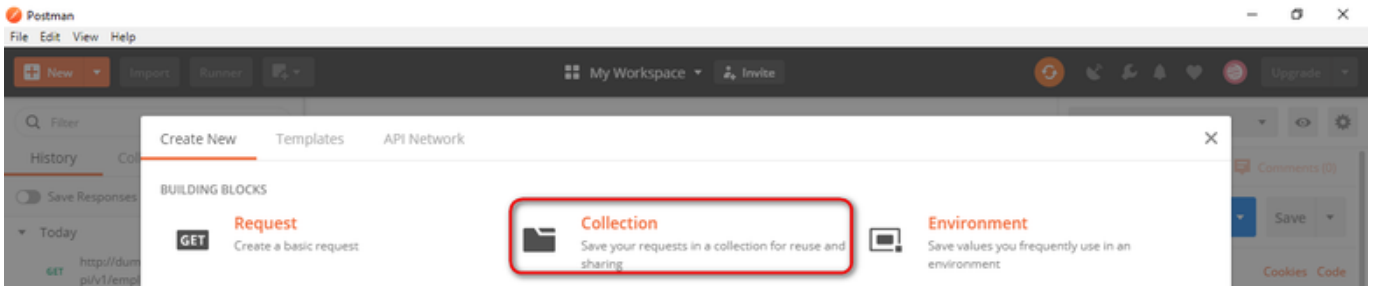
Now let's see how to create a simple request using Postman application and also see the various components of the request and its response.

2. Collections:

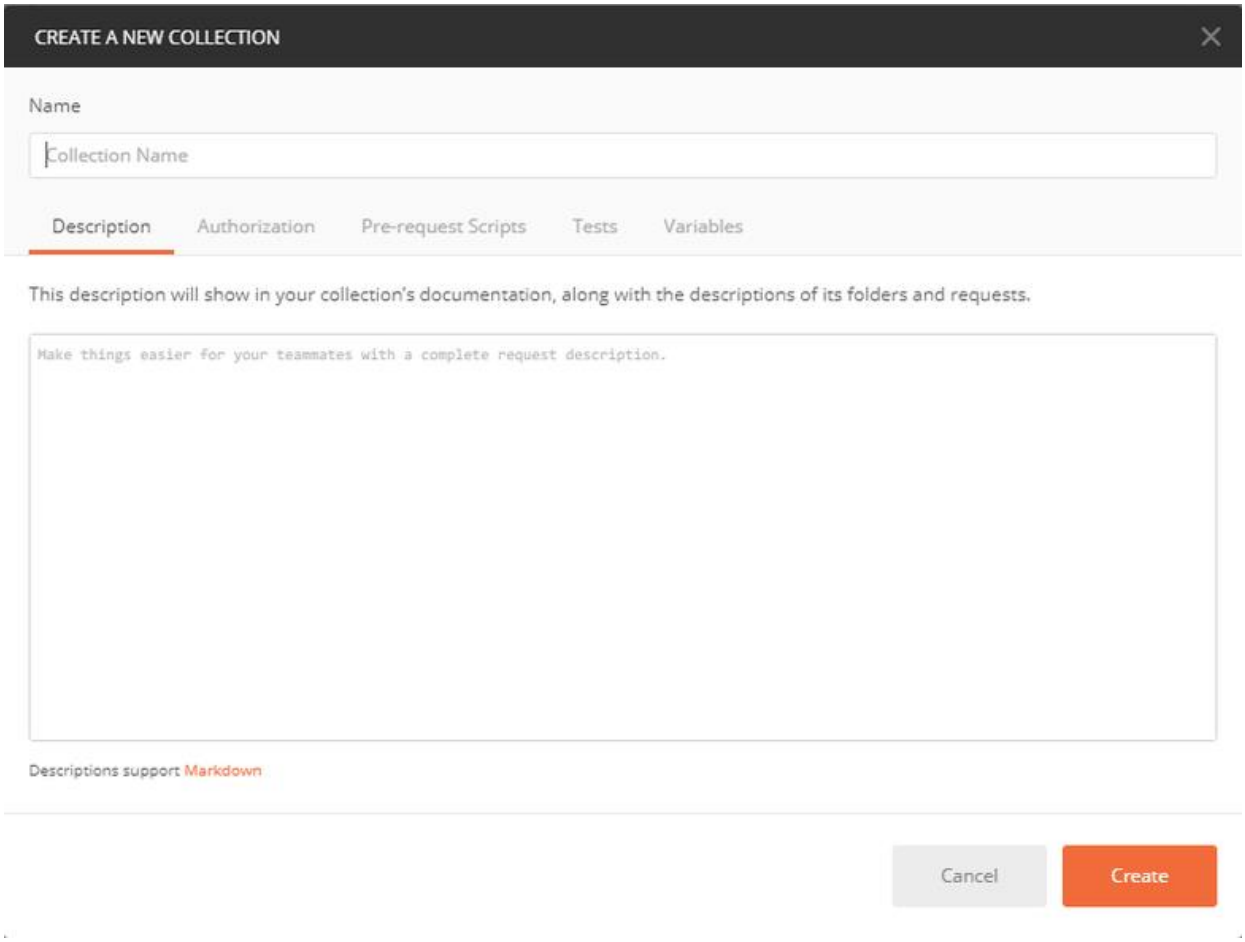
Collections are a group of saved requests you can organize into folders. We can call it as a repository to save our requests.

How To Create Collections in Postman:

Click on ***New – Collection***



Input the Collection Name and description then click Create to create a new collection



ou can add any number of requests in a Collection. You can run collections in Postman in two ways. 1. Using Collection Runner & 2. Using Newman. We will see running collections using Collection runner and Newman before closing this post.

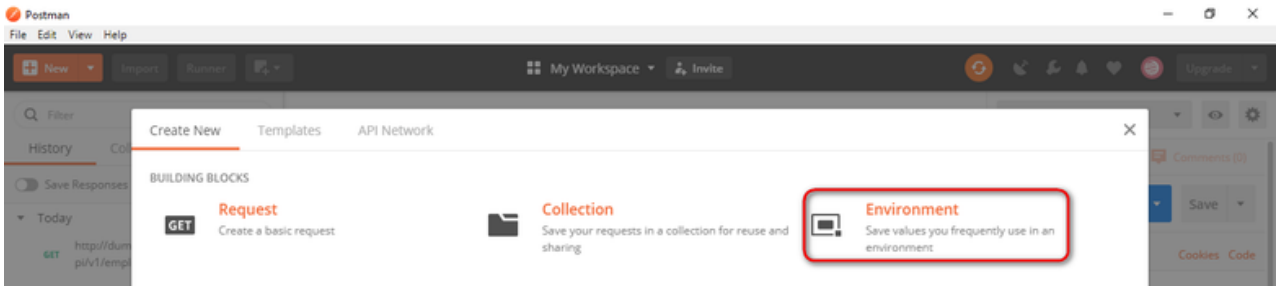
Now, let's see the third building block i.e., Environment.

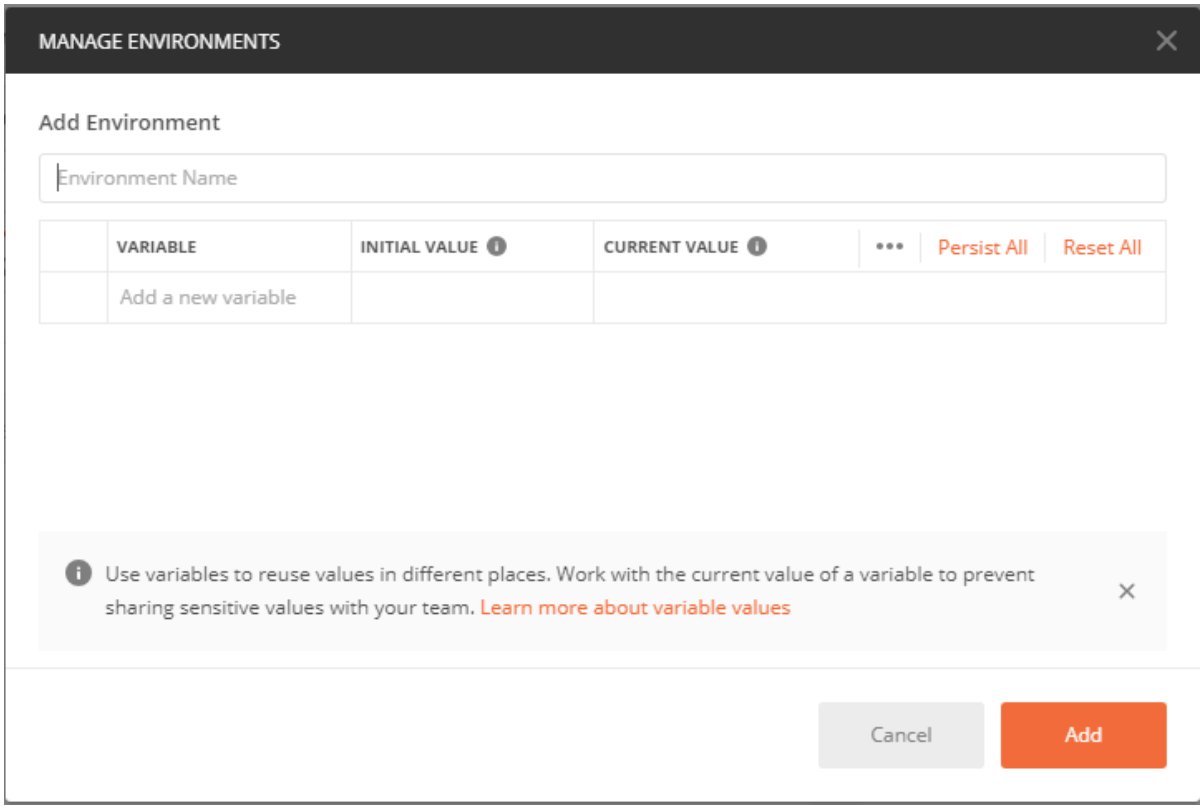
3. Environment:

Environments in Postman allow us to run requests and collections against different data sets. We could have different environments for Dev, QA & Production. Each of these environments will have different configurations such as URL, token's id and password, API keys etc., Environments are key-value pairs of variables. Each variable name represents its key. So whenever we reference a variable name then it allows us to access its corresponding value.

To create a new environment, we do as follows

Click on ***New – Environment***





Let's see how to parameterize requests after we see how to test get requests and post requests

Testing Get Requests:

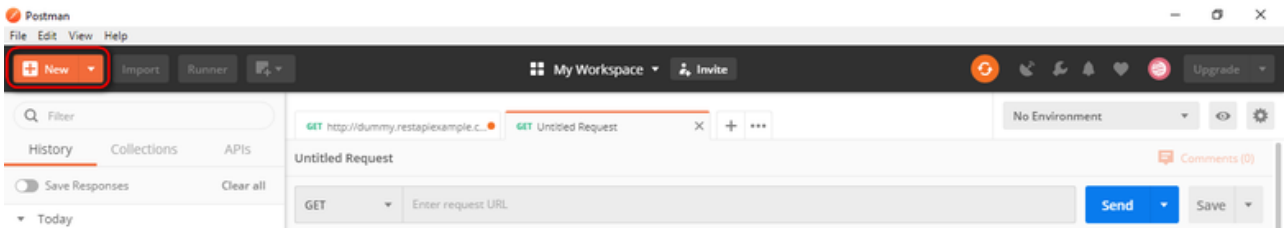
We have to use GET request to retrieve information from the given URL. With Get requests, there wont be any affect in the endpoint.

Open the Postman Application

I recommend you to sign-in to the Postman application to preserve all your actions such as requests, collections etc., for the future use. If not logged in, login with appropriate login credentials.

Here is the Postman UI initial screen.

Step 1: Click a **New** tab to create a new request.



Step 2: Creating a GET request for a REST API end point

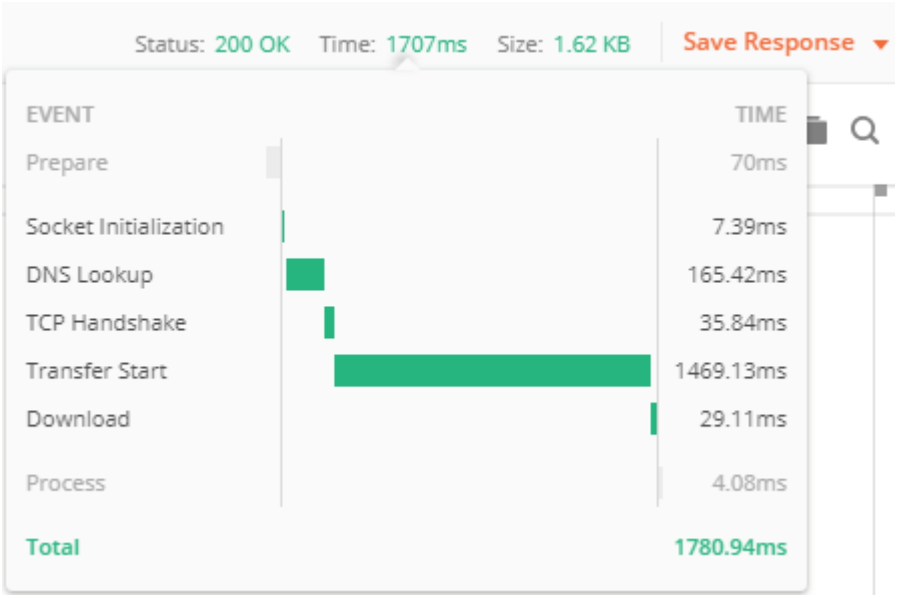
1. Set your HTTP request to **GET**
2. Input the link in request **URL** (https://jsonplaceholder.typicode.com/users)
3. Click on **SEND** to execute the request to the server hosting the endpoint
4. You can see 200 OK message in the screenshot below because our request is successful. In some cases, GET requests may be unsuccessful due to an invalid request URL or incorrect authentication.

https://jsonplaceholder.typicode.com/users
You should be able to see various data around the response after the server responds in the Body section

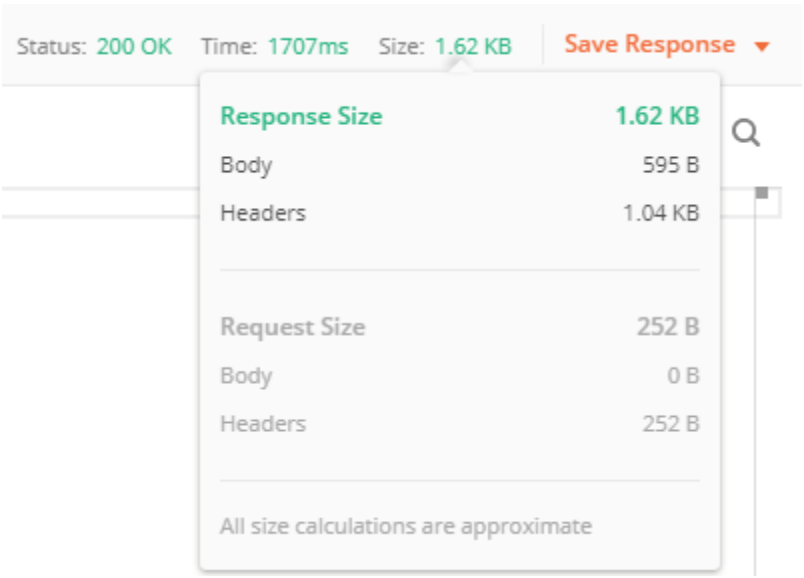
In the above screenshot under the request headers, we can see response status code, time taken for the request to complete, the size of the payload

We can find the details about the response time and response size by hover over them.

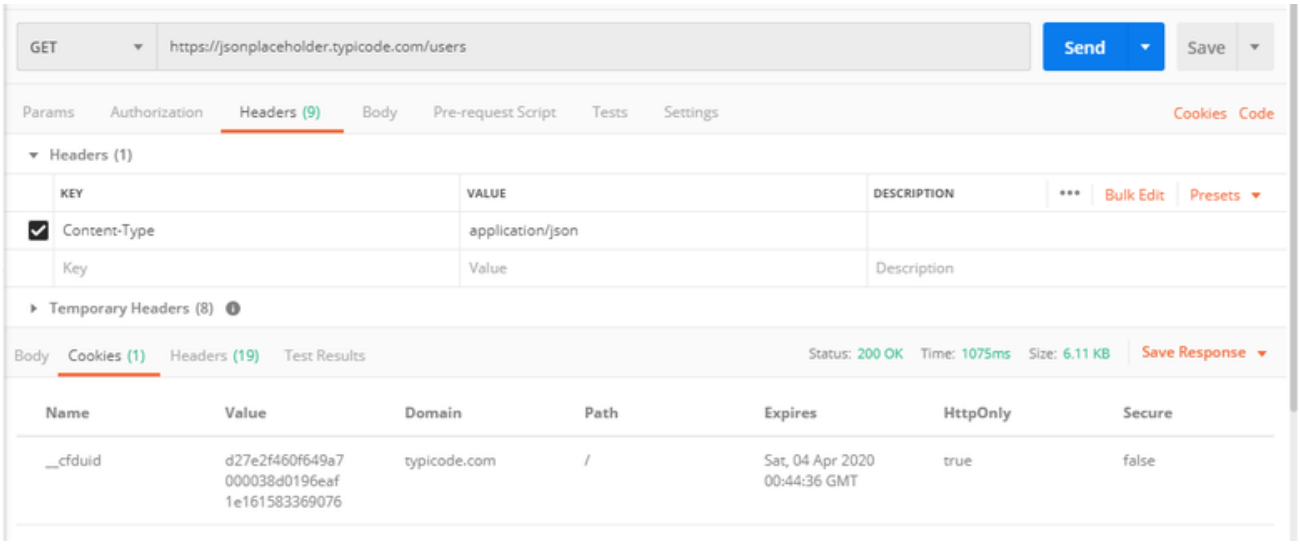
Response time: We can see individual components like Connect time, Socket time, DNS lookup, etc.,



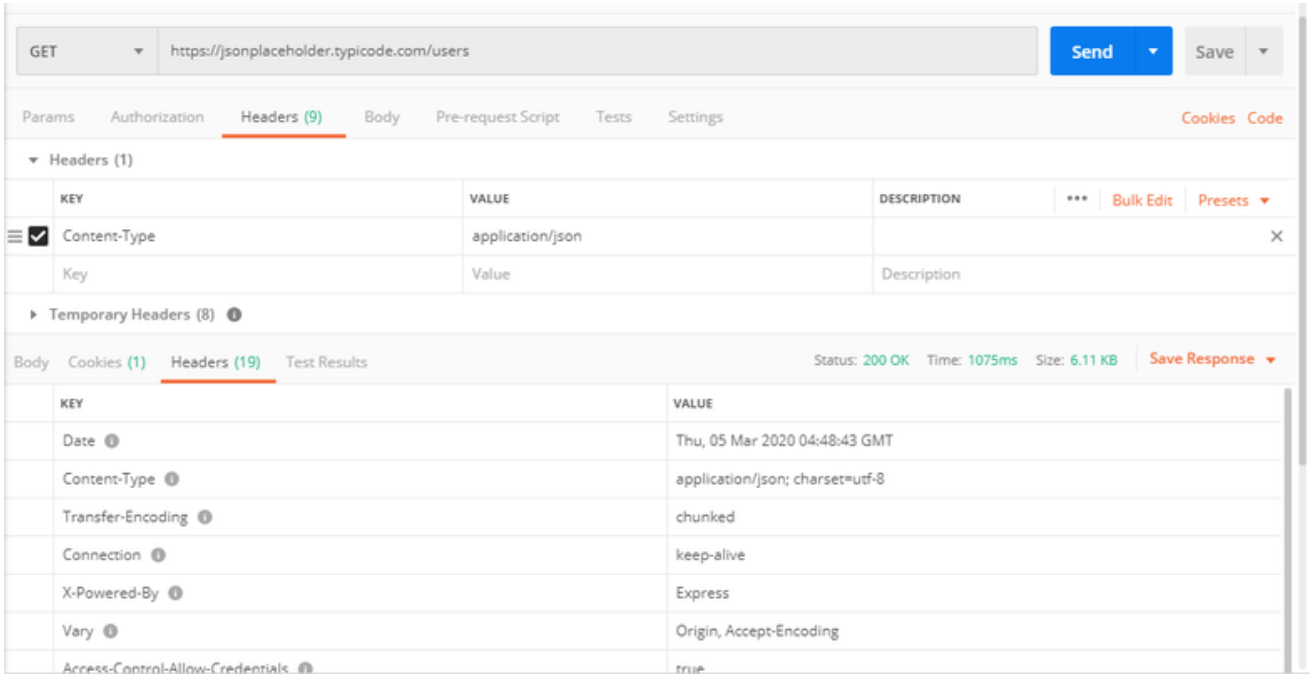
Response size: We can see individual components like actual response size, how much size the headers are constituted etc



Cookies: We can find session related information in the cookies that were returned from the server.



Response header: Here we can find information about the request that got processed.



Testing Post Requests:

Post requests are used to do data manipulation by adding data to the endpoint. Now, let's add a user into the application. To do this, we need to send data to the application. We use POST request to send data. In POST

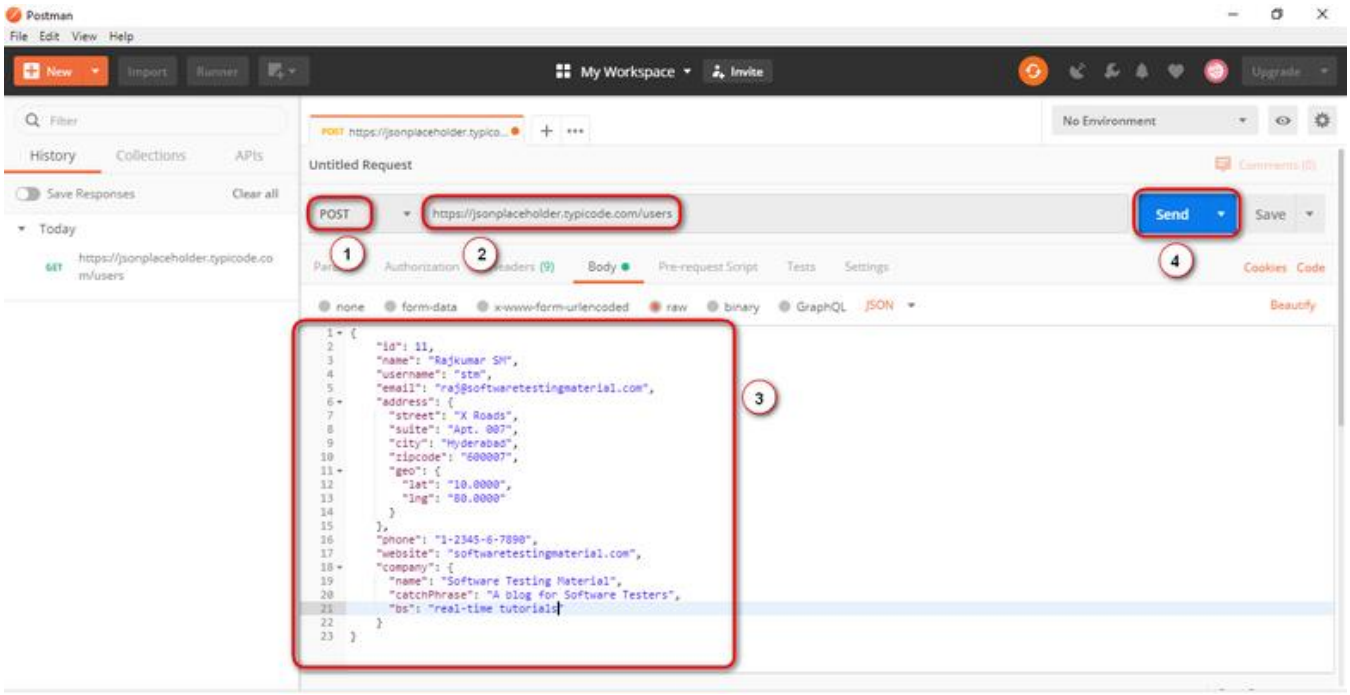
request we send data in the body of the request and API returns some data in response to the POST request to us which validates the user has been created. We use the same data which we used in GET request to add a new user.

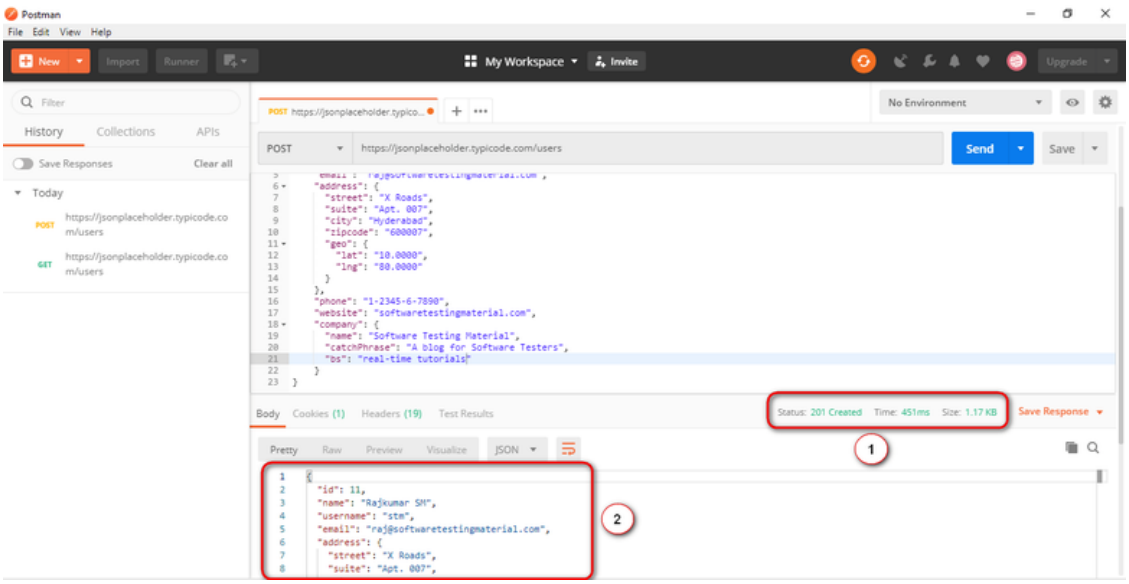
1. Set your HTTP request to **POST**
2. Input the link in request **URL** (https://jsonplaceholder.typicode.com/users)
3. Click on **Body** Tab and select "**Raw**" radio button – Select **JSON** – Copy and paste just one user result from the previous get request as shown in the below screenshot.


```
{
    "id": 11,
    "name": "Rajkumar SM",
    "username": "stm",
    "email": "raj@softwaretestingmaterial.com",
    "address": {
      "street": "X Roads",
      "suite": "Apt. 007",
      "city": "Hyderabad",
      "zipcode": "600007",
      "geo": {
        "lat": "10.0000",
        "lng": "80.0000"
      }
    },
    "phone": "1-2345-6-7890",
    "website": "softwaretestingmaterial.com",
    "company": {
      "name": "Software Testing Material",
      "catchPhrase": "A blog for Software Testers",
      "bs": "real-time tutorials"
    }
}
```


4. Click on **SEND** to execute the request to the server hosting the endpoint

Note: You can use Jsonformatter to check the correct format of the data you are trying to post.



1. You can see **201 Created message** in the screenshot below because our request is successful.
2. You can see the posted data in the body.
3. 

4. Likewise, we will test other requests PUT, PATCH & DELETE
5. **Note:** For every request, you need to check expected result, status code, response time. Also don't forgot to do negative tests to verify whether the API is responding properly or not

How To Parameterize Requests:

If we want to parameterize postman requests, we need to do as follows.

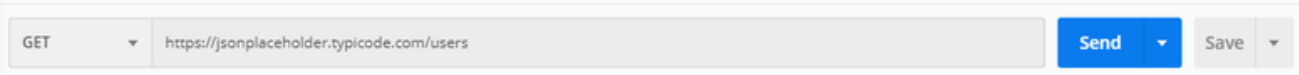
Data Parameterization is one of the most useful features of Postman.

Many times, we use same request multiple times with different data. By using Parameterization we can use variables with parameters. We can save the data in an environment variable or in a data file.

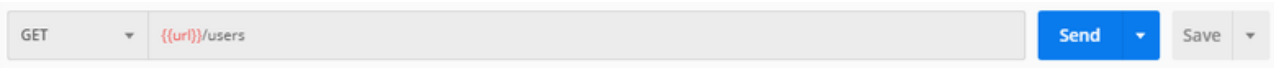
In Postman, parameters are crated through the use of double curly brackets say "{{test}}"

For example, I have a **base URL** say ***https://stm.org*** and I have stored in a variable named ***base_url***. In this case I do reference it in my requests using ***{{base_url}}***. To send a request to this base URL (ie., ***https://stm.org***) to get new customers list, I do list this base URL as part of the request URL using ***{{base_url}}/get?customers=new***. The request will be sent to ***https://stm.org/get?customers=new*** by postman.

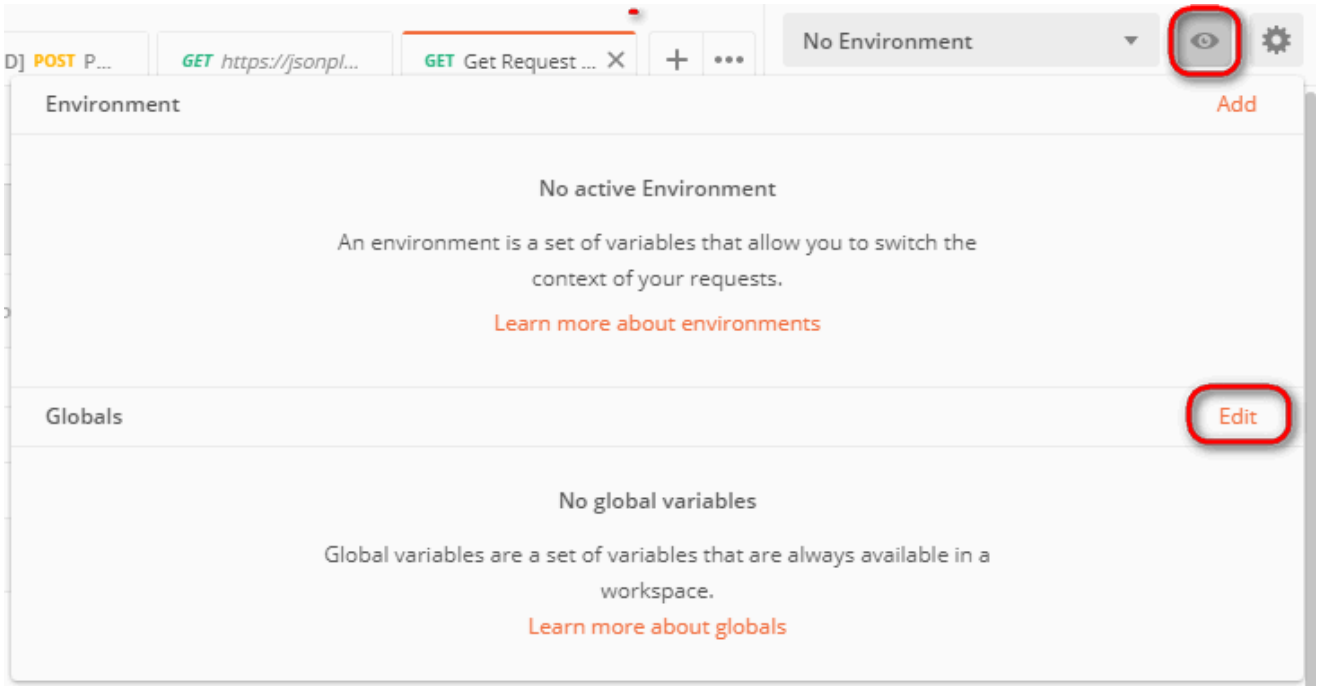
1. Set the HTTP request to GET and input the URL



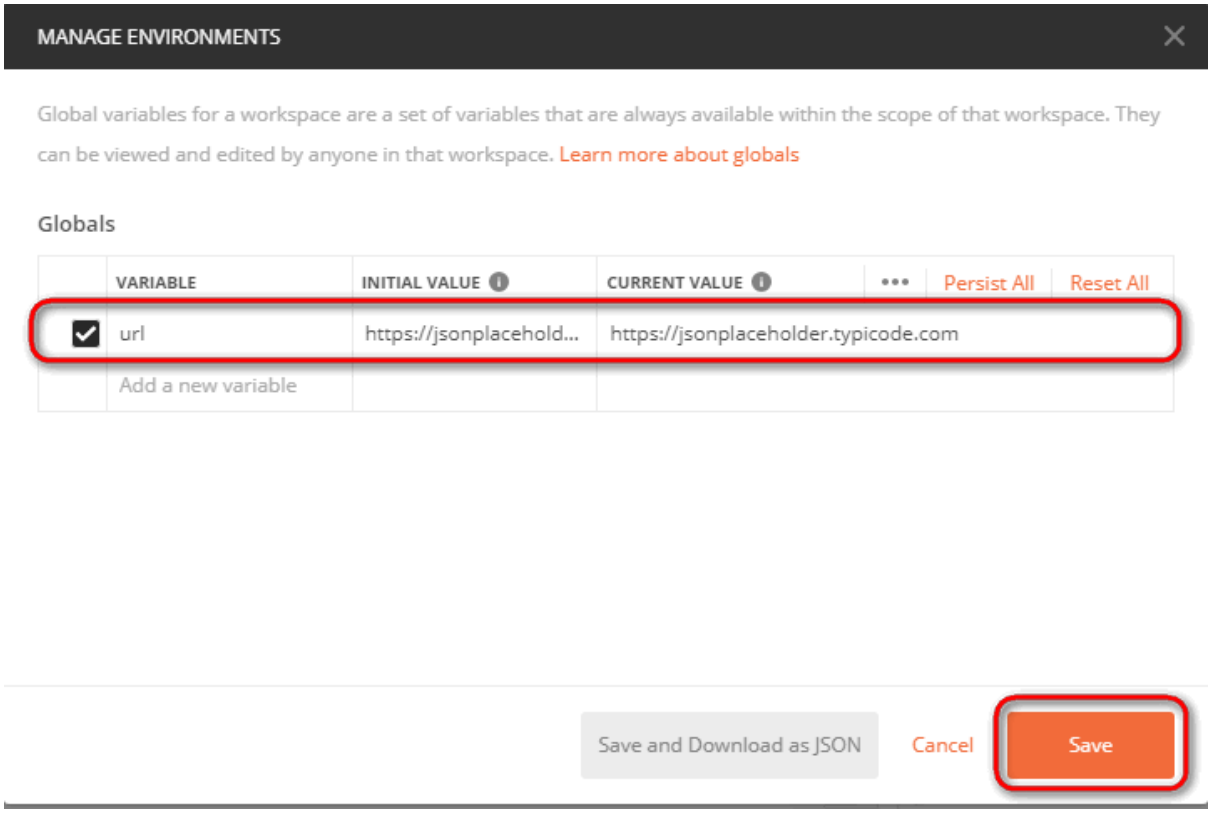
2. Replace the URL with a parameter such as ***{{url}}***. Request URL should be ***{{url}}/users***.



3. Now we need to set environment variables to set Parameters. To do click on the eye icon and click edit to set the variable to a global environment variable to use it in all collections

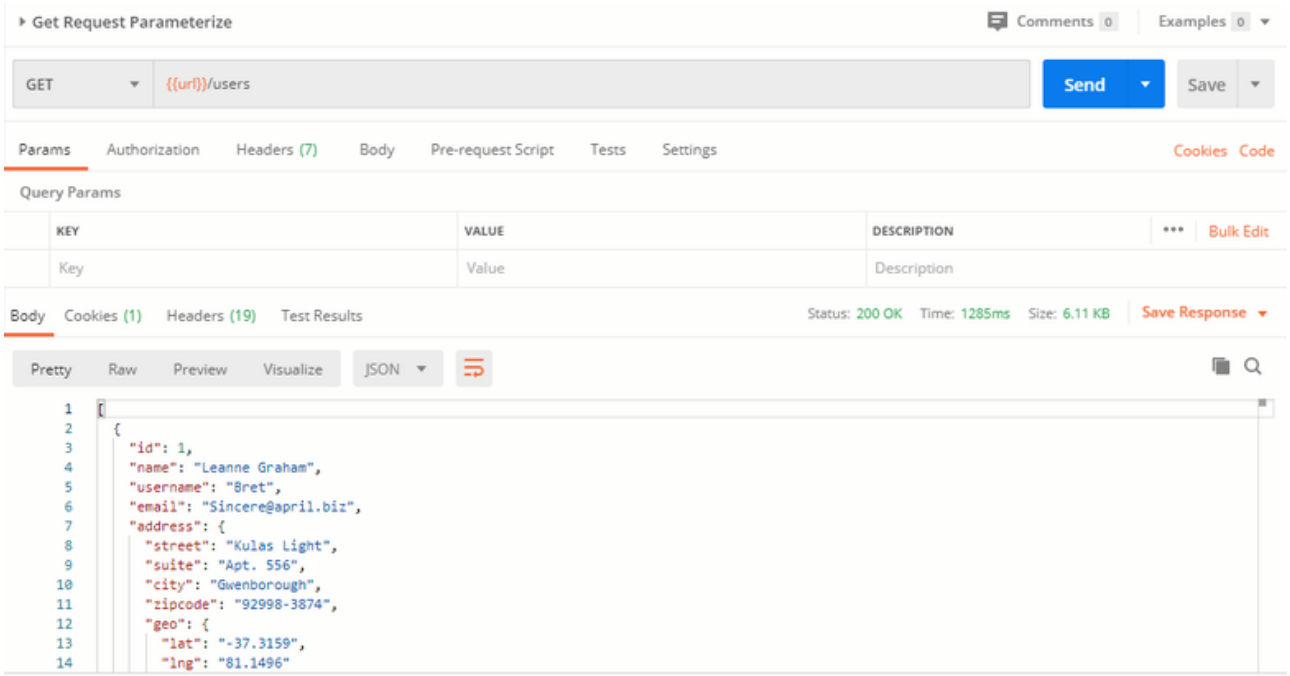


4. In variable, set the name to the url which is https://jsonplaceholder.typicode.com and click Save.



Note: Click close if you see the next screen

5. Go back to GET request and click send.

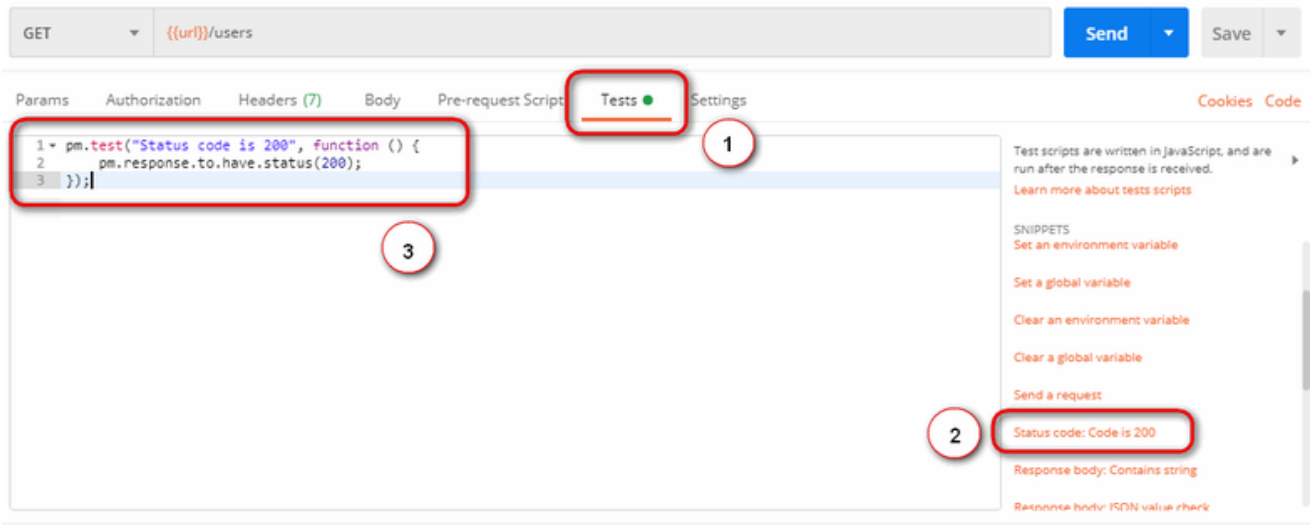


How To Create Postman Tests:

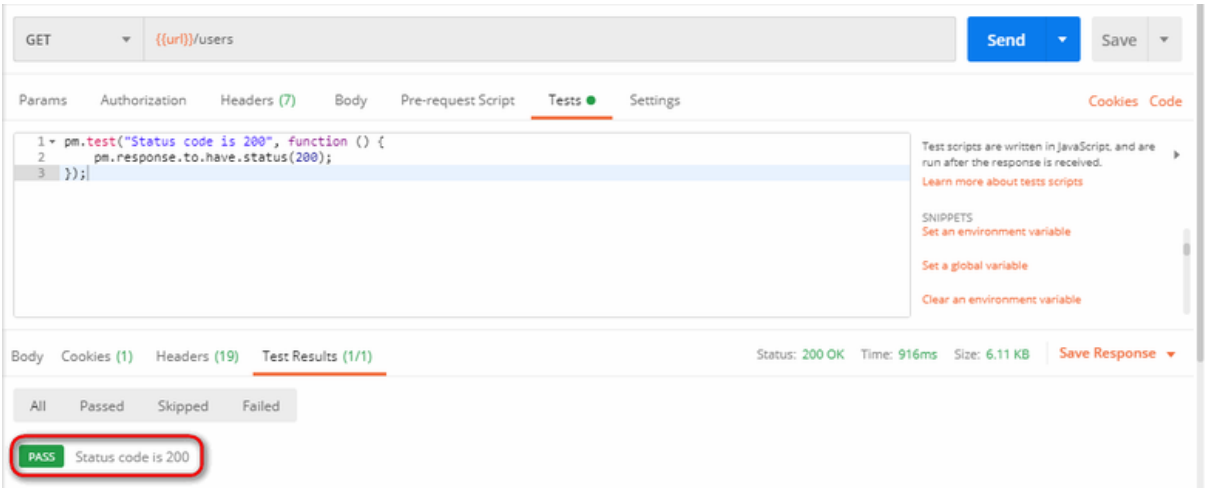
Postman Tests allow you to ensure that your API is working as expected. It is to establish integrations between services are functioning reliably, and to verify that new developments haven't broken any existing functionality. It helps you verify results such as successful or failed status, comparison of expected results etc.,

Let's start with some basic tests.

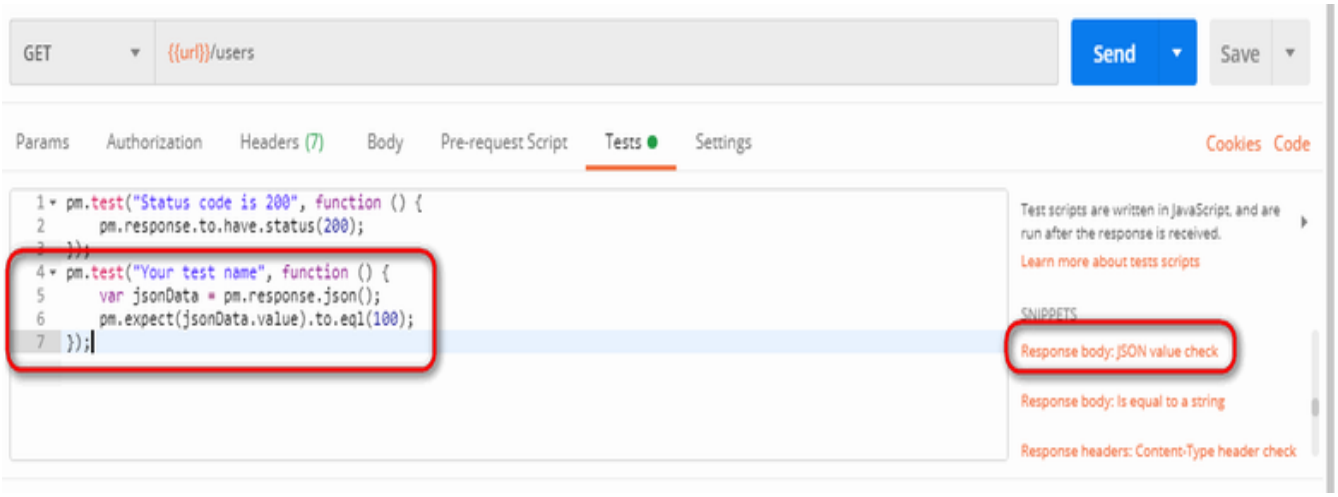
1. Go to the GET request which we created earlier. and switch to the tests tab. From the snippets section, click on "Status code: Code is 200". Script will be auto-populated.



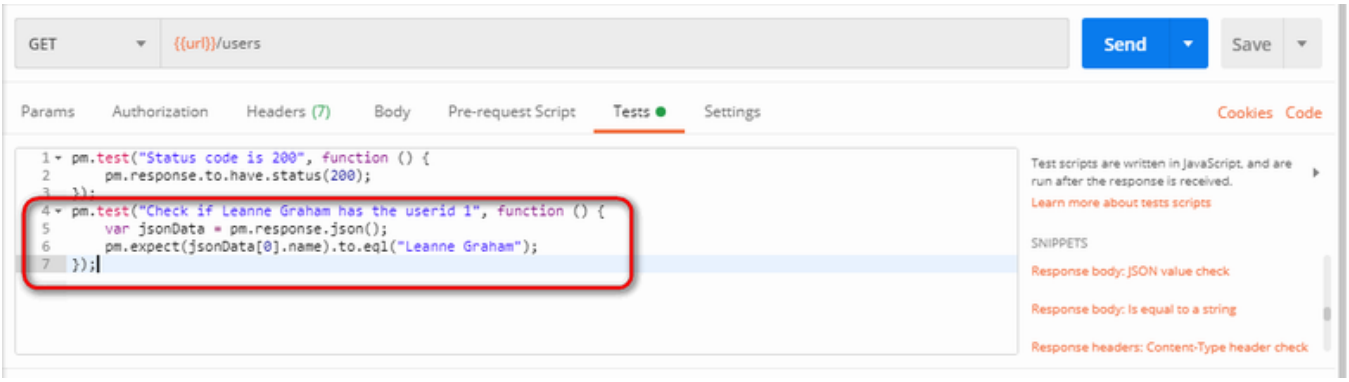
2. Click on Send. The result will be displayed.



3. Let's add another test. In this test, we do compare the expected result to the actual result. To do this, click on "Response body:JSON value check" from the snippets section. Let's check if Leanne Graham has the userid 1.

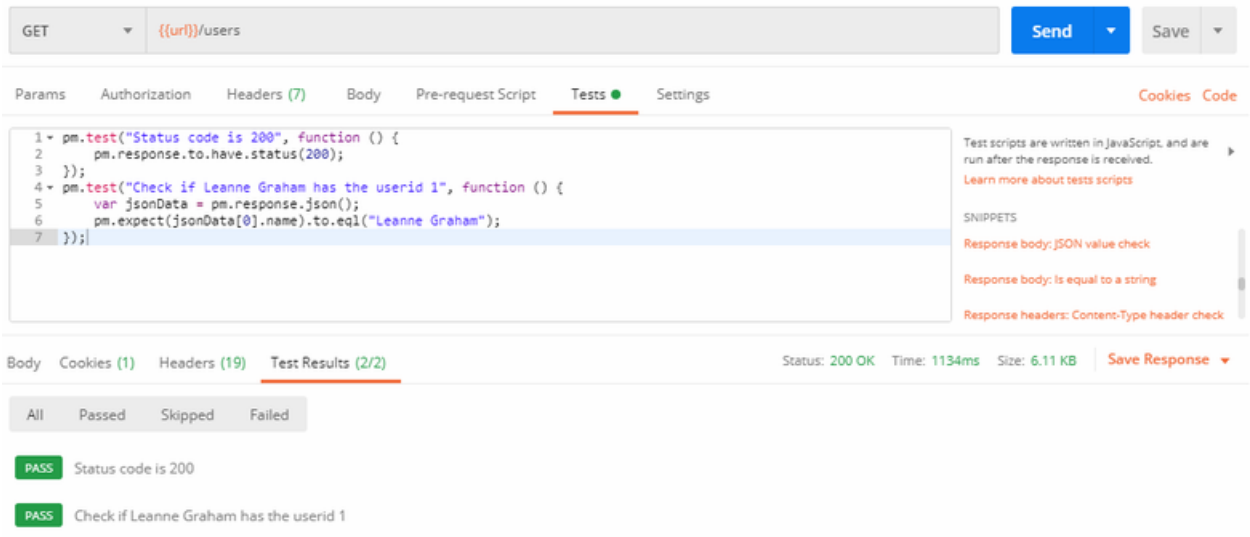


4. To specify the test name exactly what we want to test, simply replace “Your Test Name” from the code with “Check if Leanne Graham has the userid 1”. Also replace jsonData.value with jsonData[0].name. To get the path (It is there in the body of earlier GET result). Since “Leanne Graham” is userid 1, jsonData is in the first result which should start with 0. To get the second result, use jsonData[1] and so on for succeeding results.




```
pm.test("Check if user with id1 is Leanne Graham", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData[0].name).to.eql("Leanne Graham");
});
```

5. Click send.

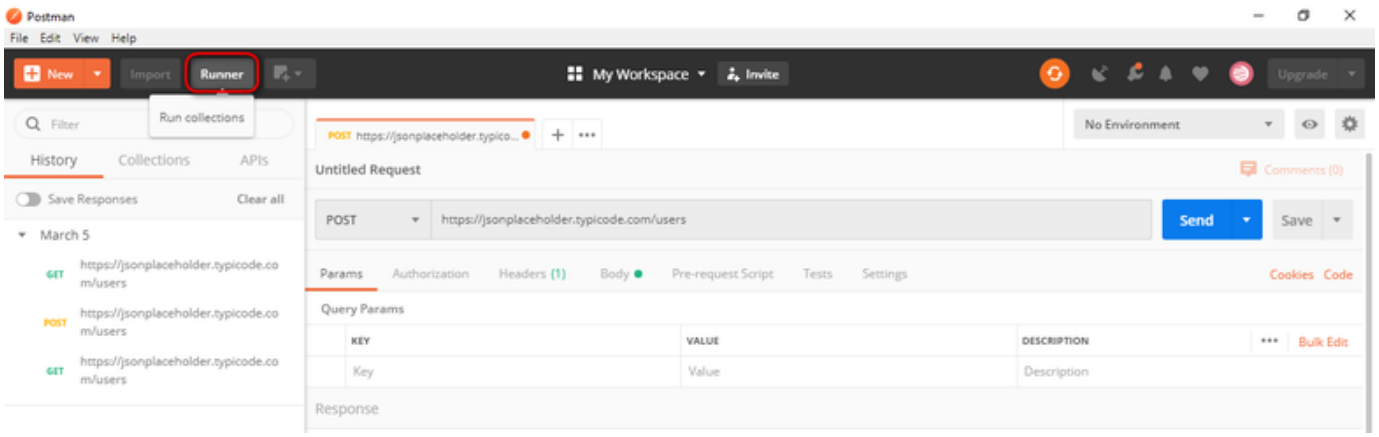


We can create tests depends on our requirement. Explore the tools by trying different tests.

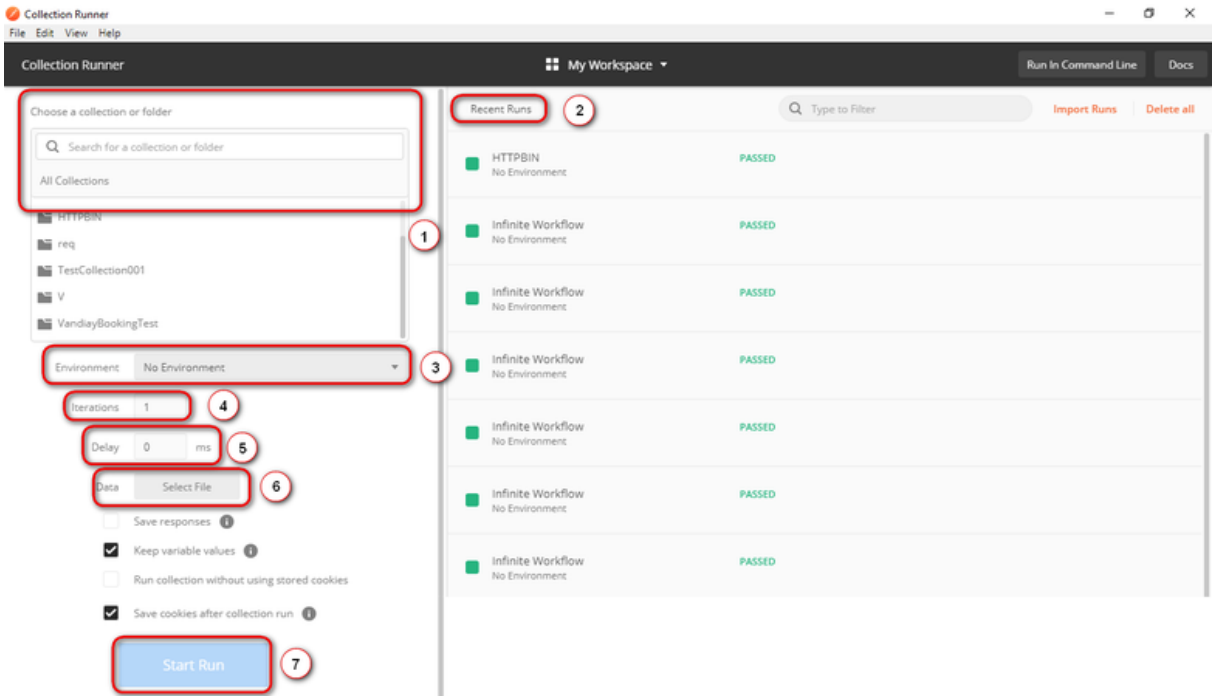
How To Run Collections using Collection Runner:

Let's run the collection using collection runner

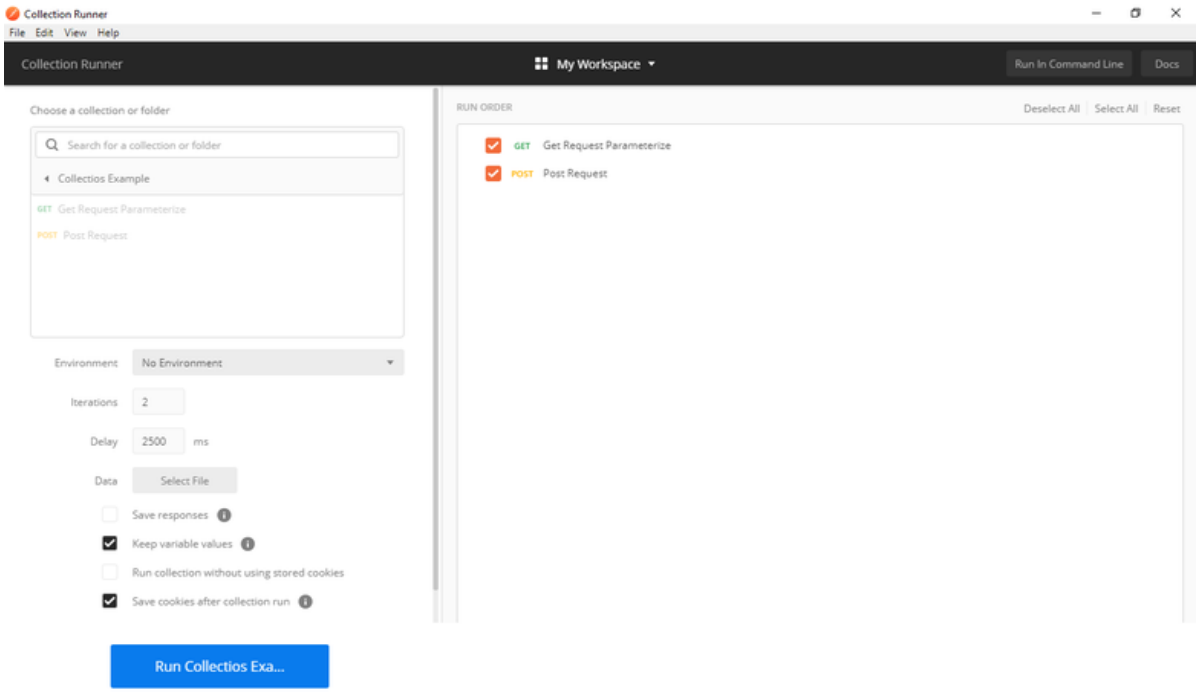
1. Click on the Runner button which is located next to the Import button.



2. Collection Runner page should appear

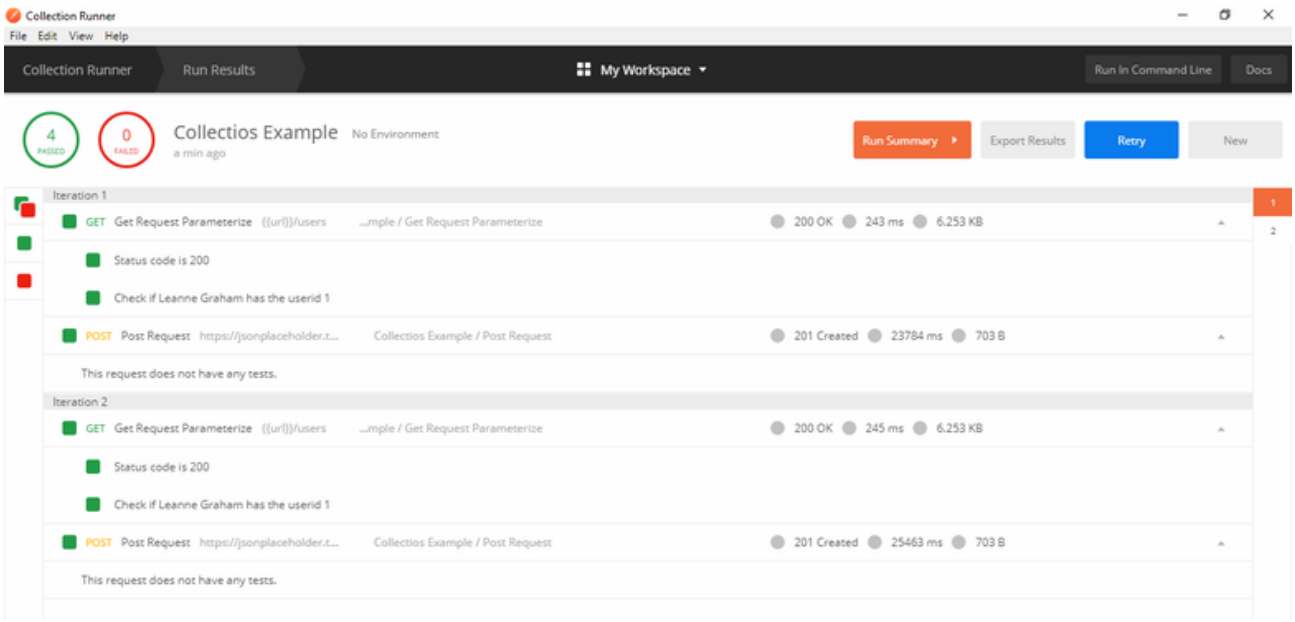


- 1. All you recent runs
- 2. If you are using specific environment then you need to select that environment
- 3. You have to set the number based on how many times you want to iterate it
- 4. Tests without delay may cause errors, so set the delay time
- 5. State how you would like to log responses
- 6. Select your data file
- 7. To run the collection
- 3. You can add select collection here. You have to select the subfolder incase of any subfolders available. Run Postman Test Collection by setting up iterations as 2, delay as 2500 ms, and click on Run Postman Test button



4. You can see the test status after the tests have executed. Here we have added both GET & POST requests but we didnt have POST requests. So you see a message for POST requests as “This request doesn’t have any tests”.

”.



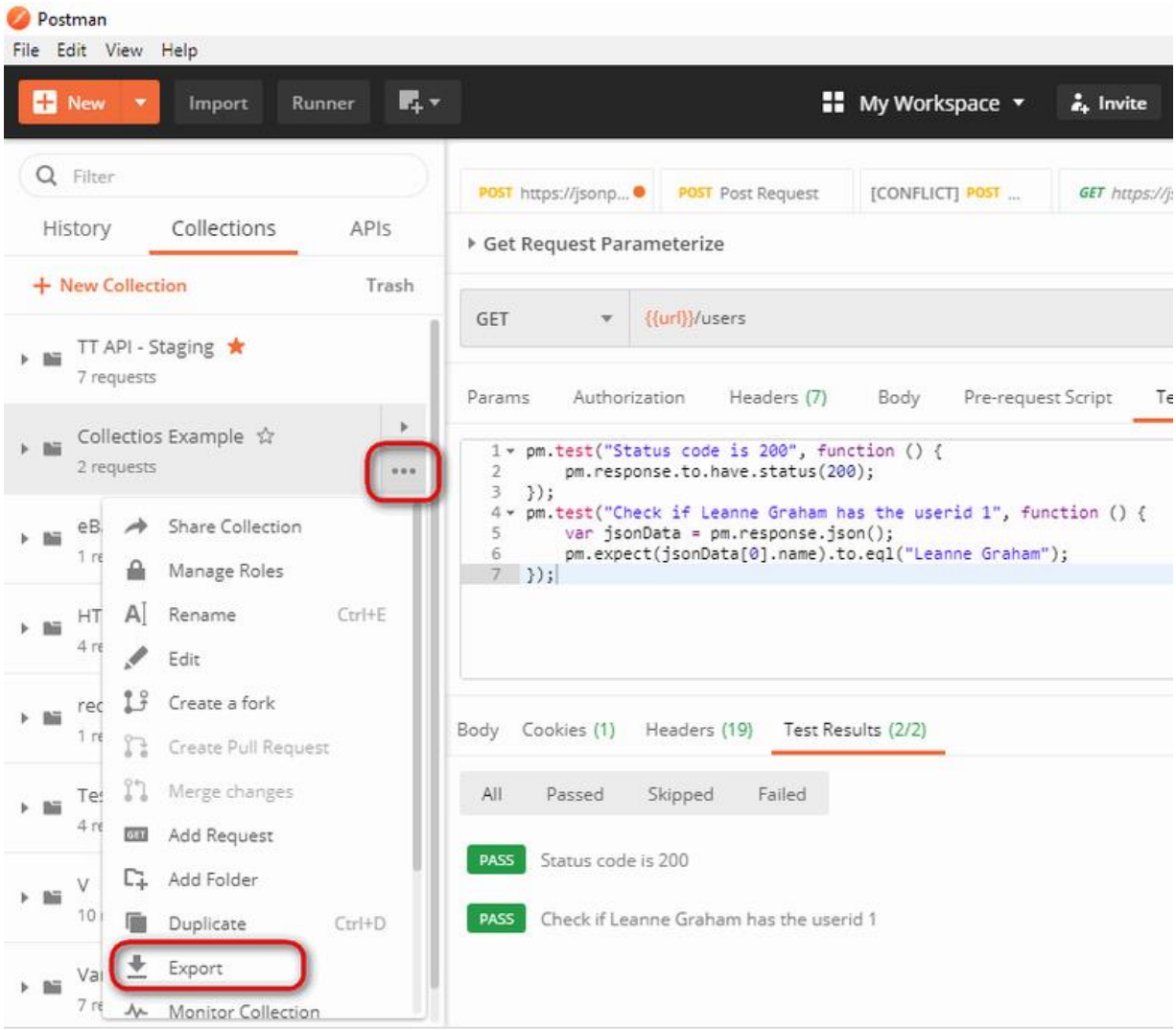
How To Run Collections using Newman:

To run our collection using Newman do the following.

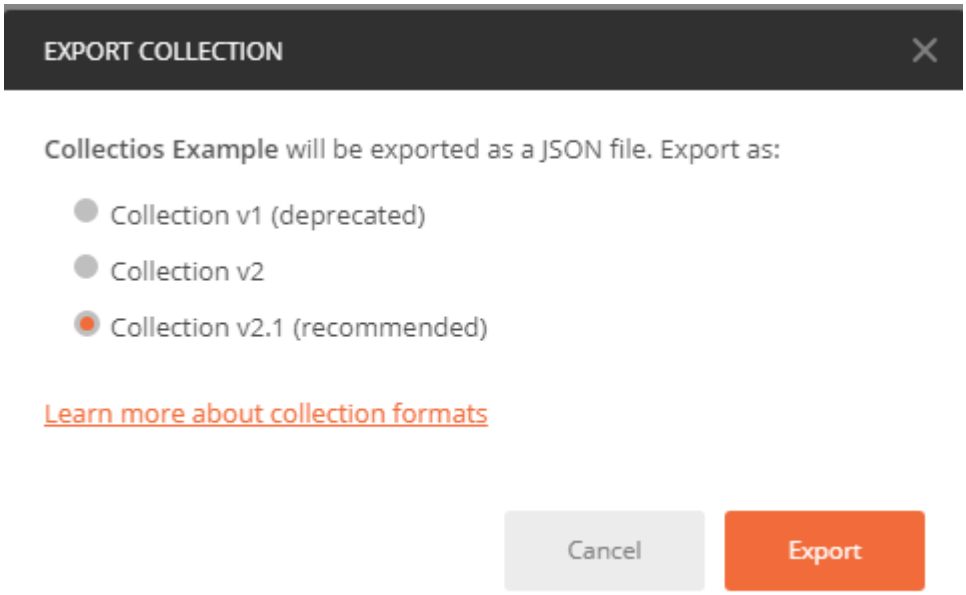
1. Install node.js using this [link]

2. Open the command line and enter ***npm install -g newman***

3. Go to the Postman workspace after the Newman is installed. In the collections box, click on the three dots and select Export.

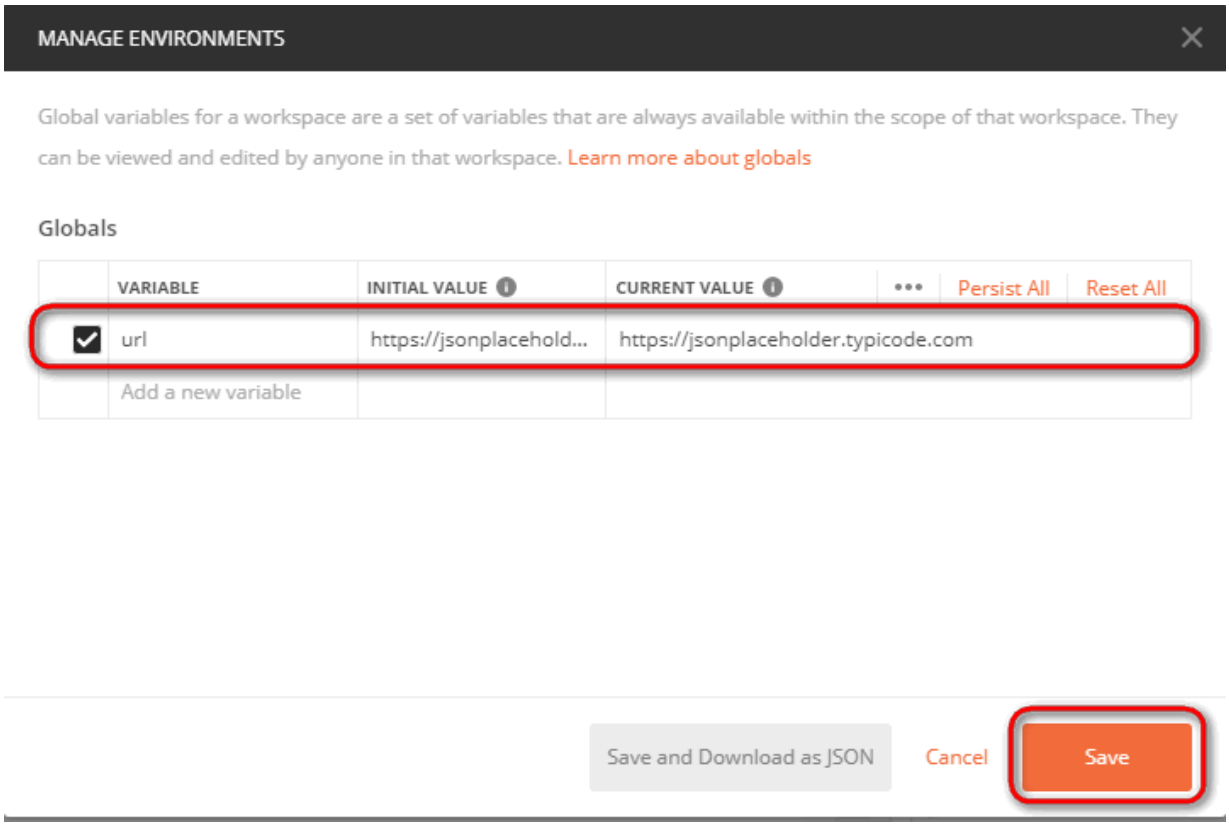


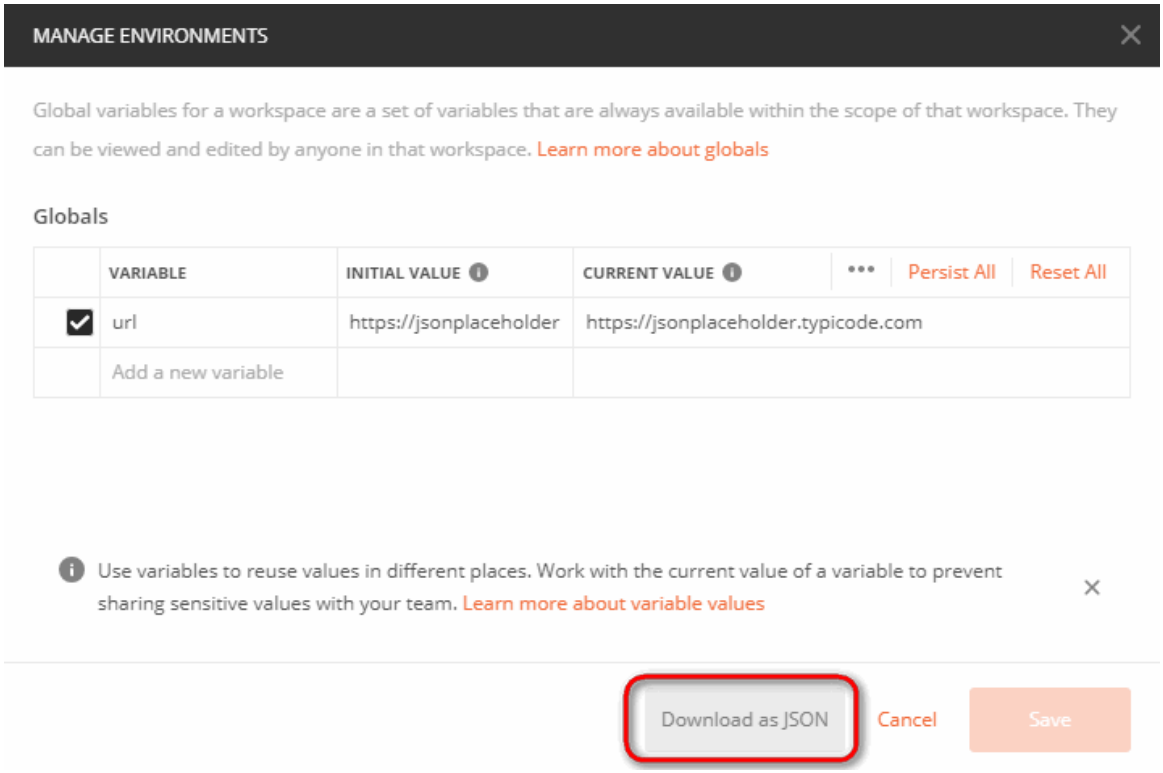
4. Choose Export Collection as Collection v2.1 (Recommended) then click Export.



5. Select your desired location then click Save. It is advisable to create a specific folder for your Postman tests. A collection should now be exported to your chosen local directory.

6. We will also need to export our environment. Click on the eye icon beside the environment dropdown in Global, select Download as JSON. Select your desired location then click Save. It is advisable that the environment should be in the same folder as your collection.





7. Environment should now be exported to the same local directory as Collection.

8. Now go back to command line and change the directory to where you have saved the collection and environment.

cd C:\Users\Asus\Desktop\Postman Tutorial

9. Run your collection using this command:

newman run PostmanTestCollection.postman_collection.json -e Testing.postman_globals.json

Run results should now appear.

API Testing Interview Questions

1. What is an API?

In simple words, API stands for **A**pplication **P**rogramming **I**nterface. API acts as an interface between two software applications and allows the two software applications to communicate with each other. API is a collection of software functions that can be executed by another software program

API is an acronym and it stands for **A**pplication **P**rogramming **I**nterface. API is a set of routines, protocols, and tools for building Software Applications. APIs specify how one software program should interact with other software programs.

2. What is API Testing?

API testing is a type of software testing that involves testing APIs directly and also as a part of integration testing to check whether the API meets expectations in terms of functionality, reliability, performance, and security of an application. In API Testing our main focus will be on the Business logic layer of the software architecture. API testing can be performed on any software system which contains multiple APIs.

3. What are the common API Testing Types?

API testing typically involves the following practices:

- Unit testing
- Functional testing
- Load testing
- Runtime/ Error Detection
- Security testing
- UI testing
- Interoperability and WS Compliance testing
- Penetration testing
- Fuzz testing

4. Name some of the common protocols used in API Testing?

Some of the protocols using in API Testing are as follows:

- HTTP
- REST
- SOAP
- JMS
- UDDI

5. What are some of the architectural styles for creating a Web API?

Some of the architectural styles for creating web api are as follows.

- Simple URI as the address for the services
- Stateless communication
- HTTP for client-server communication
- XML/JSON as formatting language

6. What is API test environment?

In API testing environment, no GUI (Graphical User Interface) is available.

For API, the test environment is a rather sophisticated approach that configures the server and database to match the requirement of the software application. After the installation process is done, API will be verified for correct functioning.

Throughout the process, various parameters for the original environment are established via API calls to examine the test results.

7. Difference between API and Web services?

Web services:

1. All web services are APIs
2. All web services need to be exposed over web(HTTP)
3. A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication
4. A Web service always needs a network to operate

APIs:

1. All APIs are not web services
2. All APIs need not be exposed over web(i.e. HTTP)
3. API uses multiple ways for communication e.g. DLL files in C/C++, Jar files/ RMI in java, Interrupts in Linux kernel API etc.
4. APIs don't need a network for operation

8. What is Soap?

SOAP stands for Simple Object Access Protocol. It is an XML based messaging protocol. It helps in exchanging information among computers

9. What is Rest API?

REST stands for Representational State Transfer. It is a set of functions helping developers in performing requests and receive responses. Interaction is made through HTTP Protocol in REST API.

10. Difference between SOAP and REST?

SOAP:
1. SOAP is a protocol through which two computers communicate by sharing XML document
2. SOAP supports only XML format
3. SOAP does not support caching
4. SOAP is slower than REST
5. SOAP is like a custom desktop application, closely connected to the server
6. SOAP runs on HTTP but envelopes the message

REST:
1. REST is a service architecture and design for network-based software architecture
2. REST supports different data formats
3. REST supports caching
4. REST is faster than SOAP
5. REST client is just like a browser and uses standard methods An application has to fit inside it
6. REST uses the HTTP headers to hold meta information

11. What are the common tests that are performed on APIs?

Some of the common tests we perform on APIs are as follows.

1. Verify whether the return value is based on the input condition. The response of the APIs should be verified based on the request.
2. Verify whether the system is authenticating the outcome when the API is updating any data structure
3. Verify whether the API triggers some other event or request another API
4. Verify the behavior of the API when there is no return value

12. What are the advantages of API Testing?

- API Testing is time effective when compared to GUI Testing. API test automation requires less code so it can provide faster and better test coverage.
- API Testing helps us to reduce the testing cost. With API Testing we can find minor bugs before the GUI Testing. These minor bugs will become bigger during GUI Testing. So finding those bugs in the API Testing will be cost-effective to the Company.
- API Testing is language independent.
- API Testing is quite helpful in testing Core Functionality. We can test the APIs without a user interface. In GUI Testing, we need to wait until the application is available to test the core functionalities.
- API Testing helps us to reduce the risks.

13. What exactly needs to be verified in API Testing?

Basically, on API Testing, we send a request to the API with the known data and we analyze the response.
1. Data accuracy
2. HTTP status codes
3. Response time
4. Error codes in case API return any errors
5. Authorization checks
6. Non-functional testing such as performance testing, security testing

14. Name some tools used for API Testing?

Some of the tools used for API Testing are as follows:

Some of the tools used for API Testing are as follows:

- Postman
- Katalon Studio
- SoapUI
- Assertible
- Tricentis Tosca
- Apigee
- JMeter
- Rest-Assured
- Karate DSL
- API Fortress
- Parasoft
- HP QTP(UFT)
- vREST
- Airborne
- API Science
- APIary Inspector
- Citrus Framework
- Hippie-Swagger
- HttpMaster Express
- Mockbin
- Ping API
- Pyresttest

- Rest Console
- RoboHydra Server
- SOAP Sonar
- Unirest
- WebInject

15. List some most used templates for API documentation?

Some of the API documentation templates are as follows.

- Swagger
- FlatDoc
- RestDoc
- API blueprint
- Slate
- Miredot
- Web service API Specification.

16 Name some of the API examples which are quite popular.

Some of the popular API examples are

- Google Maps API
- YouTube
- Twitter
- Amazon Advertising API

17. Difference between API testing and Unit Testing?

UNIT TESTING:

- Unit testing is conducted by the Development Team
- Unit testing is a form of White box testing

- Unit testing is conducted prior to the process of including the code in the build
- Source code is involved in Unit testing
- In unit testing, the scope of testing is limited, so only basic functionalities are considered for testing

API TESTING:

- API testing is conducted by QA Team
- API testing is a form of Black box testing
- API testing is conducted after the build is ready for testing
- Source code is not involved in API testing
- In API testing, the scope of testing is wide, so all the issues that are functional are considered for testing

18. What are the main challenges faced in API testing?

Some of the challenges we face while doing API testing are as follows

- Selecting proper parameters and its combinations
- Categorizing the parameters properly
- Proper call sequencing is required as this may lead to inadequate coverage in testing
- Verifying and validating the output
- Due to the absence of GUI, it is quite difficult to provide input values

19. What are the types of bugs we face when performing API testing?

Issues observed when performing API testing are

- Stress, performance, and security issues
- Duplicate or missing functionality
- Reliability issues
- Improper messaging
- Incompatible error handling mechanism
- Multi-threaded issues
- Improper errors

20. How is UI testing is not similar to API testing?

UI (User Interface) testing is to test the graphical interface part of the application. Its main focus is to test the look and feel of an application. On the other hand, API testing enables the communication between two different software systems. Its main focus is in the business layer of the application.

21. Name some most commonly used HTTP methods?

Some of the HTTP methods are

GET: It enables you to retrieve data from a server
POST: It enables you to add data to an existing file or resource in a server
PUT: It lets you replace an existing file or resource in a server
DELETE: It lets you delete data from a server
PATCH: It is used to apply partial modifications to a resource
OPTIONS: It is used to describe the communication options for the target resource
HEAD: It asks for a response identical to that of a GET request, but without the response body

22. Can you use GET request instead of PUT to create a resource?

No, GET request only allows read only rights. It enables you to retrieve data from a server but not create a resource. PUT or POST methods should be used to create a resource.

23. What is the difference between PUT and POST methods?

PUT and POST methods are sometimes confused in regards to when each should be used. Using POST request, our intent is to create a new object on the server whereas with PUT request, our intent is to replace an object by another object.

POST should be used when the client sends the page to the server and then the server lets the client know where it put it. PUT should be used when the client specifies the location of the page

Method in REST. •

Difference between PUT and PATCH call

• How to integrate postman to project? •

How will you handle dynamic payloads in API?

• How do you capture specific responses value and pass to other request?

• What challenges you faced in API testing?

• What is difference between Authorization and Authentication?

• What are the API status codes, you have come across?

• What is difference between OAuth1.0 and OAuth2.O ,When and where do you use and how. Can you write a sample code?

• How you get the response from one api and send to another api?

SOAP Interview Questions

1. What are SOAP Web services?

SOAP is defined as an XML-based protocol. It is known for designing and developing web services as well as enabling communication between applications developed on different platforms using various programming languages over the Internet. It is both platform and language independent.

2. How does SOAP work?

SOAP is used to provide a user interface that can be accessed by the client object, and the request that it sends goes to the server, which can be accessed using the server object. The user interface creates some files or methods consisting of server object and the name of the interface to the server object. It also contains other information such as the name of the interface and methods. It uses HTTP to send the XML to the server using the POST method, which analyzes the method and sends the result to the client. The server creates more XML consisting of responses to the request of user interface using HTTP. The client can use any approach to send the XML, like the SMTP server or POP3 protocol to pass the messages or reply to queries.

3. When to use SOAP API?

Use the SOAP API to create, retrieve, update or delete records, like accounts, leads, and use-defined objects. With more than 20 different calls, you can also use the SOAP API to manage passwords, perform searches, etc. by using the SOAP API in any language that supports web services.

4. How users utilize the facilities provided by SOAP?

PutAddress(): It is used to enter an address in the webpage and has an address instance on the SOAP call.
PutListing(): It is used to allow the insertion of a complete XML document into the web page. It receives the XML file as an argument and transports the XML file to XML parser liaison, which reads it and inserts it into the SOAP call as a parameter.
GetAddress(): It is used to get a query name and gets the result that best matches a query. The name is sent to the SOAP call in the form of text character string.
GetAllListing(): It is used to return the full list in an XML format.

5. What is the major obstacle users faced when using SOAP?

When using SOAP, users often see the firewall security mechanism as the biggest obstacle. This block all the ports leaving few like HTTP port 80 and the HTTP port used by SOAP that bypasses the firewall. The technical complaint against SOAP is that it mixes the specification for message transport with the specification for message structure.

6. What are the various approaches available for developing SOAP based web services?

There are two different methods available for developing SOAP-based web services, which are explained below:
Contract-first approach: the contract is first defined by XML and WSDL, and then Java classes are derived from the contract.
Contract-last approach: Java classes are first defined, and then the contract is

generated, which is normally the WSDL file from the Java class.
“Contract-first” method is the most popular approach.

7. What are the elements of a SOAP message structure?

It is a common XML document that contains the elements as a SOAP message

Envelope: It is an obligatory root element that translates the XML document and defines the beginning and end of the message.
Header: It is an optional item which contains information about the message being sent.
Body: It contains the XML data comprising the message being sent.
Fault: It provides the information on errors that occurred while during message processing.

8. What are the syntax rules for a SOAP message?

- Must use encoded XML
- Envelope namespace must be used
- Encoding namespace must be used
- Must not consist of a DTD reference
- Must not have XML processing instruction

9. What is the transport method in SOAP?

Application layer and transport layers of a network are used by SOAP; HTTP and SMTP are the valid protocol of the application layer used as the transport for SOAP. HTTP is more preferable, since it works well with the current Internet infrastructure, in particular with firewalls.
The SOAP requests can be sent using an HTTP GET method while the specification only contains details about HTTP POST.

10. What are some important characteristics of a SOAP envelope element?

- SOAP message has a root Envelope element

- Envelope is an obligatory part of the SOAP message.
- If an envelope includes a header element, it should not contain more than one.
- Envelop version will change if the SOAP version changes.
- The SOAP envelope is indicated by the prefix ENV and the envelope element.
- The optional SOAP encoding is also specified using a namespace and the optional encoding style element.

11. What are the major functionalities provided by the SOAP protocol class?

The SOAP protocol is used to provide simple access methods for all the applications available on the Internet, providing the following functionalities:

- Call: A class which provides the main functionality for a remote method for which a call is needed. It is used to create the call() and to specify the encoding style of the registry that will be assigned when if necessary. This call() function is used by the RPC call, which represents the options of the call object.
- Deployment Descriptor: A class used to provide the information about the SOAP services. It enables easy deployment without the need for other approaches.
- DOM2 Writer: A class that serializes and uses DOM node as XML string to provide more functionalities.
- RPC Message: A class used as the base class that calls and replies to the request submitted to the server.
- Service Manager: A class that provides, lists and then outputs all SOAP services.

12. What are the web relation functionalities provided by SOAP protocol?

- HTTPUtils: This provides the functionality of the POST method to safely meet the requirements.
- Parameter: It is an argument for a RPC call used by both the client and the server.

- Response: It is an object that represents an RPC reply from both client and server, but the result will not be displayed until after the method call.
- TCPTunnel: It is an object that provides the ability to listen on a specific port and to forward all the host and port names.
- TypeConverter: It helps to convert an object of one type into another type and this is called using the class in the form object.

13. How does the message security model allow the creation of SOAP more secure to use?

The security model includes the given security tokens. These tokens comprise digital signatures for protection and authentication of SOAP messages. Security tokens can be used to provide the bond between authentication secrets or keys and security identities. Security token uses the authentication protocols and an X.509 certificate to define the relationship between the public key and identity key. The signatures are used to verify the messages and their origin, generate knowledge to confirm the security tokens to bind the identity of a person to the identity of the originator. Security model prevents different attacks and can be used to protect the SOAP architecture.

14. What is the difference between top down & bottom up approach in SOAP Web services?

- Top down SOAP Web services include creating WSDL document to create a contract between the web service and the client, with a required code as an option. This is also known as Contract-first approach. The top-down approach is difficult to implement because classes must be written to confirm the contract defined in WSDL. One of the benefits of this method is that both client and server code can be written in parallel.
- Bottom up SOAP web services require the code to be written first and then WSDL is generated. It is also known as Contract-last approach. Since WSDL is created based on the code, bottom-up approach is easy to implement and client codes must wait for WSDL from the server side to start working.

15. What are advantages of SOAP?

- SOAP is both platform and language independent.
- SOAP separates the encoding and communications protocol from the runtime – environment.
- Web service can retrieve or receive a SOAP user data from a remote service, and the source's platform information is completely independent of each other.
- Everything can generate XML, from Perl scripts through C++ code to J2EE app servers.
- It uses XML to send and receive messages.
- It uses standard internet HTTP protocol.
- SOAP runs over HTTP; it eliminates firewall problems. When protocol HTTP is used as the protocol binding, an RPC call will be automatically assigned to an HTTP request, and the RPC response will be assigned to an HTTP reply.
- Compared to RMI, CORBA and DCOM, SOAP is very easy to use.
- SOAP acts as a protocol to move information in a distributed and decentralized environment.
- SOAP is independent of the transport protocol and can be used to coordinate different protocols.

16. What are disadvantages of SOAP?

SOAP is typically significantly slower than other types of middleware standards, including CORBA, because SOAP uses a detailed XML format. A complete understanding of the performance limitations before building applications around SOAP is hence required.
SOAP is usually limited to pooling and not to event notifications when HTTP is used for the transport. In addition, only one client can use the services of one server in typical situations.
If HTTP is used as the transport protocol, firewall latency usually occurs since the firewall analyzes the HTTP transport. This is because HTTP is also leveraged for Web browsing, and so many firewalls do not understand the difference between using HTTP within a web browser and using HTTP within SOAP.
SOAP has different support levels, depending on the supported programming language. For instance, SOAP supported in Python and PHP is not as powerful as it is in Java and .NET.

17. SOAP or Rest APIs, which method to use?

SOAP is the heavyweight choice for Web service access. It provides the following advantages when compared to REST:

- SOAP is not very easy to implement and requires more bandwidth and resources.
- SOAP message request is processed slower as compared to REST and it does not use web caching mechanism.
- WS-Security: While SOAP supports SSL (just like REST) it also supports WS-Security which adds some enterprise security features.
- WS-AtomicTransaction: Need ACID Transactions over a service, you're going to need SOAP.
- WS-ReliableMessaging: If your application needs Asynchronous processing and a guaranteed level of reliability and security. Rest doesn't have a standard messaging system and expects clients to deal with communication failures by retrying.
- If the security is a major concern and the resources are not limited then we should use SOAP web services. Like if we are creating a web service for payment gateways, financial and telecommunication related work, then we should go with SOAP as here high security is needed.

REST is easier to use for the most part and is more flexible. It has the following advantages when compared to SOAP:

- Since REST uses standard HTTP, it is much simpler.
- REST is easier to implement, requires less bandwidth and resources.
- REST permits many different data formats whereas SOAP only permits XML.
- REST allows better support for browser clients due to its support for JSON.
- REST has better performance and scalability. REST reads can be cached, SOAP based reads cannot be cached.
- If security is not a major concern and we have limited resources. Or we want to create an API that will be easily used by other developers publicly then we should go with REST.
- If we need Stateless CRUD operations then go with REST.
- REST is commonly used in social media, web chat, mobile services and Public APIs like Google Maps.
- RESTful service returns various MediaTypes for the same resource, depending on the request header parameter "Accept" as

application/xml or application/json for POST and /user/1234.json or GET /user/1234.xml for GET.
- REST services are meant to be called by the client-side application and not the end user directly.
- ST in REST comes from State Transfer. You transfer the state around instead of having the server store it, this makes REST services scalable.

18. What are the factors that help to decide which style of Web services – SOAP or REST – to use?

Generally, REST is preferred due to its simplicity, performance, scalability, and support for multiple data formats.
However, SOAP is favorable to use where service requires an advanced level of security and transactional reliability.
But you can read the following facts before opting for any of the styles.

- Does the service expose data or business logic? REST is commonly used for exposing data while SOAP for logic.
- Requirement from clients or providers for a formal contract. SOAP can provide contract via WSDL.
- Support multiple data formats.
- Support for AJAX calls. REST can apply the XMLHttpRequest.
- Synchronous and asynchronous calls. SOAP enables both synchronous/ asynchronous operations whereas REST has built-in support for synchronous.
- Stateless or Stateful calls. REST is suited for stateless operations.
- Security. SOAP provides a high level of security.
- Transaction support. SOAP is good at transaction management.
- Limited bandwidth. SOAP has a lot of overhead when sending/receiving packets since it's XML based, requires a SOAP header. However, REST requires less bandwidth to send requests to the server. Its messages are mostly built using JSON.
- Ease of use. REST based application is easy to implement, test, and maintain.

Test Automation Framework Interview Questions And Answers:

1. What is a Framework*?*

A framework defines a set of rules or best practices which we can follow in a systematic way to achieve the desired results.

So the above-mentioned test automation frameworks deal with best practices to achieve the goals of our automation project.

Let's see a general example:

Most of us love a cup of tea. How we make a good tea.

To make it, we add ingredients like tea powder, sugar, milk and water to make a tea. To make a good tea, all the ingredients we add should be in right ratio.

In case you want to make it on daily basis how you do. Its not possible to add all the ingredients in right ratio on everyday.

If you add all the ingredients in right ratio in a jar. You can make the tea everytime with same taste.

Here the '**jar**' where we added all the ingredients required to make a good tea

What is Selenium Framework?

Selenium framework's code structure helps you to reuse the code, reduce code maintenance, higher code readability, and allows multiple users to work on the same piece of the program

Why do we need the Selenium Framework?

- Easy code maintenance
- Increase in code re-usage
- Higher code readability
- Reduced script maintenance cost
- Reduced tests' time execution
- Reduced human resources

- Easy reporting

2. Tell me some popular Test Automation Frameworks?

There are different types of test automation frameworks and the most common ones are:

- Modular Testing Framework
- Data Driven Testing Framework
- Keyword Driven Testing Framework
- Hybrid Testing Framework
- Behavior Driven Development Framework

Linear Scripting Framework:

Linear Scripting Framework is a basic level test automation framework that is in the form of 'Record and Playback' in a linear fashion.

This framework is also known as the 'Record and Playback' framework.

This type of framework is used to test small-sized applications.

In this type, the creation, and execution of test scripts are done individually for each test case individually.

Testers capture each test step such as browsing, navigation, user inputs, enforcing checkpoints. Testers then play the scripts to carry out the tests.

Advantages of Linear Scripting Automation Framework:

- Can generate test scripts (Record and playback) without planning much or consume much time
- Coding knowledge is not required
- A quick way to generate test scripts

Disadvantages of Linear Scripting Automation Framework:

- Lack of reusability due to autogenerated scripts
- Hard coding the data doesn't allow us to run with multiple data sets
- Maintenance is high – It requires a lot of effort to do even small changes.

Modular Testing Framework:

In the modular testing framework, testers create test scripts module wise by breaking down the complete application under test into smaller, independent tests.

In simple words, testers divide the application into multiple modules and create test scripts individually. These individual test scripts can be combined to make larger test scripts by using a master script to achieve the required scenarios. This master script is used to invoke the individual modules to run end to end test scenarios.

The main reason for using this framework is to build an abstraction layer to safeguard the master module from any changes made in individual tests.

In this framework, testers write function libraries to use it whenever required. This is AKA modularity framework or module-based framework.

Advantages of Modular Testing Framework:

- Better scalability and easier to maintain due to breaking down the complete application into different modules
- Can write test scripts independently
- Changes in one module bring no or low impact on the other modules

Disadvantages of Modular Testing Framework:

- Takes more time to analyze the test cases and to identify reusable flows
- Due to hardcoded data in the test scripts, it's not possible to sue multiple data sets.
- Requires coding skills to set up the framework

Library Architecture Testing Framework:

Library Architecture Testing framework aka “Structured Scripting” or “Functional Decomposition”

It is based on the modular framework with some additional advantages.

In the modular testing framework, we divide the application under test into modules whereas here we identify the common tasks and grouped them into functions. Once the functions are grouped then these groups will be kept in a library. The test scripts reuse these libraries to create new test cases.

Advantages of a LIbrary Architecture Testing Framework:

- Script maintenance is simple
- Easy to scalable
- Functions library is reusable and it can be reusable

Disadvantages of a LIbrary Architecture Testing Framework:

- Coding skills are required
- It takes more time to prepare test scripts
- A fixed set of test data is hardcoded within the scripts

Data-driven Framework:

The data-driven test automation framework is focused on separating the test scripts logic and the test data from each other.

It allows us to create test automation scripts by passing different sets of test data.

The test data set is kept in the external files or resources such as MS Excel Sheets, MS Access Tables, SQL Database, XML files, etc.,

The test scripts connect to the external resources to get the test data.

By using this framework we could easily make the test scripts work properly for different sets of test data.

This framework significantly reduces the number of test scripts compared to the module-based framework.

This framework gives more test coverage with reusable tests and flexibility in the execution of tests only when required and by changing only the input test data.

It is reliable in terms of no impact on tests by changing the test data but it has its own drawbacks such as testers who work on this framework needs to have the hands-on programming knowledge to develop test scripts

Advantages of a Data-Driven Framework:

- It supports multiple data sets
- Modifying the test scripts won't affect the test data
- No need to hardcode test data
- Saves time by executing more tests

Disadvantages of a Data-Driven Framework:

- Require coding skills
- Setting up the framework and test data takes more time
- Need experienced automation testers to design framework

Keyword Driven Testing Framework:

It is also known as table-driven testing or action word based testing.

In Keyword-driven testing, we use a table format to define keywords or action words for each function or method that we would execute.

It performs automation test scripts based on the keywords specified in the excel sheet.

By using this Framework, testers can work with keywords to develop any test automation script, testers with less programming knowledge would also be able to work on the test scripts.

The logic to read keywords and call the required action mentioned in the external excel sheet is placed in the main class. Keyword-driven testing is similar to data-driven testing.

Even though to work on this framework doesn't require much programming skills but the initial setup (implement the framework) requires more expertise.

Advantages of Keyword-Driven Frameworks:

- No need to be an expert to write test scripts
- It is possible to reuse the code. We can point the different scripts to the same keyword
- Even though application changes, test scripts don't change.
- Tests can be designed before developing the application
- Test scripts work independently of an application under test with basic modifications
- Not dependent on test tools

Disadvantages of Keyword-Driven Frameworks:

- Take more time to design
- The initial cost is high
- Employees with good test automation skills needed

Hybrid Driven Testing Framework:

Hybrid Test automation framework is the combination of two or more frameworks mentioned above. It attempts to leverage the strengths and benefits of other frameworks for the particular test environment it manages. Most of the teams are building this hybrid driven framework in the current market.

Behavior Driven Development Testing Framework:

The purpose of this Behavior Driven Development framework is to create a platform that allows everyone (such as Business Analysts, Developers, Testers, etc,) to participate actively. It requires increased collaboration between Development and Test Teams. It doesn't require the users to be acquainted with a programming language. We use non-technical, natural language to create test specifications. Some of the tools available in the market for Behavior Driven Development is JBehave, Cucumber, etc.,

The frameworks stated above are some of the most popular **Test Automation Frameworks** used by the automation testers.

3. Why Framework?

In a test automation project, we do perform different tasks by using different types of files. To organize and manage all the files and to finish all the tasks in a systematic approach we use a framework.

4. Have you created any Framework?

If you are a beginner: *No, I didn't get a chance to create a framework. I have used the framework which is already available.*
If you are an experienced tester: *Yes, I have created a framework (Or) No, but I have involved in the creation of the framework.*

5. What are the advantages of using Test Automation Framework?

1. Saves time and money. Automation testing is faster in execution
2. Reusability of code. Create one time and execute multiple times with less or no maintenance
3. Easy reporting. It generates automatic reports after test execution
4. Easy for compatibility testing. It enables parallel execution in combination of different OS and browser environments
5. Low cost maintenance. It is cheaper compared to manual testing in a long run
6. Automated testing is more reliable
7. Automated testing is more powerful and versatile
8. It is mostly used for regression testing. Supports execution of repeated test cases
9. Minimal manual intervention. Test scripts can be run unattended
10. Maximum coverage. It helps to increase the test coverage

6. Which Test Automation Framework you are using and why?

Some of the Test Automation Frameworks are:

- Data Driven Testing Framework
- Keyword Driven Testing Framework
- Hybrid Testing Framework

7. Mention the name of the framework which 'you are currently using' or which 'you have hands on experience'.

Example:
Answers should be, *Already the organization which I am working for is using that particular framework* or *I have an experience on that particular framework* or *It's easy to handle all my scripts to execute and generate logs, screenshots and reports* by using this framework.

13. What type of test cases do you pick up to automate?

I focus on the test cases which should be executed in a repetitive manner such as regression test cases, smoke and sanity test cases

14. What type of test cases you won't pick up to automate?

Before picking up the test cases to automate, I do check whether the application is stable or not. So based on this, I don't pickup test cases when the AUT changes frequently and the test cases which I run rarely and run only one time. When I do usability and exploratory testing.

15. How many test cases you have automated per day?

It depends on Test case scenario complexity and length. I did automate 2-5 test scenarios per day when the complexity is limited. Sometimes just 1 or fewer test scenarios in a day when the complexity is high.

16. How you build Object Repository in your project?

In QTP, there is an Object Repository concept. When a user records a test, the objects and its properties are captured by default in an Object Repository. QTP uses this Object Repository to play back the scripts. Coming to Selenium, there is no default Object Repository concept. It doesn't mean that there is no Object Repository in Selenium. Even though there is no default one still we could create our own. In Selenium, we call objects as locators (such as ID, Name, Class Name, Tag Name, Link Text, Partial Link Text, XPath, and CSS). Object repository is a collection of objects. One of the ways to create Object Repository is to place all the locators in a separate file (i.e., properties file). But the best way is to use Page Object Model. In the Page Object Model Design Pattern, each web page is represented as a class. All the objects related to a particular page of a web application are stored in a class.

Maven:

What is Maven?

Maven is a popular open-source build tool developed by the Apache Group to build, publish, and deploy several projects at once for better project management. The tool provides allows developers to build and document the lifecycle framework.

Maven is written in Java and is used to build projects written in C#, Scala, Ruby, etc. Based on the Project Object Model (POM), this tool has made the lives of Java developers easier while developing reports, checks build and testing automation setups.

Maven focuses on the simplification and standardization of the building process, taking care of the following:

- Builds
- Documentation
- Dependencies
- Reports
- SCMs
- Distribution
- Releases
- Mailing list

• Lifecycle of Maven

The Maven build lifecycle is a sequence of phases that are executed sequentially to deploy and distribute a project:

- Validate: Checks that the project is correct and has all the necessary information
- Compile: Compiles the project's source code
- Test: Tests the compiled source code using a unit testing framework
- Package: Packages the compiled code into a distributable format, such as a JAR or WAR
- Integration test: Processes and deploys the package into an environment for integration tests

- Verify: Runs checks to ensure the package is valid and meets quality standards
- Install: Installs the package into the local repository so other projects can use it as a dependency
- Deploy: Copies the packaged code to a remote repository for sharing with other developers

• Use of Maven surfire plugin. If yes, where and why?

What is the use of pom.xml?

POM is an acronym for Project Object Model. The pom. xml file contains information of project and configuration information for the maven to build the project such as dependencies, build directory, source directory, test source directory, plugin, goals etc. Maven reads the pom.

CI / CD tools • What is Jenkins?

Jenkins is an open-source automation server that's commonly used for continuous integration (CI) and continuous delivery (CD) in software development. It's a Java-based program that can be used on Windows, Linux, macOS, and other Unix-like operating systems

How will you handle dependencies in Maven at run time?

Maven does so by reading project files (pom. xml) of dependencies, figure out their dependencies and so on. We only need to define direct dependency in each project pom. Maven handles the rest automatically.

Can you give some basic commands used in maven project?

Certainly! Here are some basic Maven commands you'll find useful for managing your projects:

1. **Create a New Maven Project**:
   - To create a new Maven project, use the following command:
   - `mvn archetype:generate -DgroupId=com.example -DartifactId=myproject -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false`

    - Replace `com.example` with your desired group ID and `myproject` with your project's name.
2. **Build the Project**:
    - To compile your project and generate the output (usually JAR or WAR files), use:
    - `mvn clean install`
3. **Run Tests**:
    - Execute unit tests using:
    - `mvn test`
4. **Clean the Project**:
    - To remove generated files (like compiled classes), use:
    - `mvn clean`
5. **Update Dependencies**:
    - To update dependencies in your `pom.xml`, use:
    - `mvn versions:display-dependency-updates`
6. **Generate Site Documentation**:
    - To create project documentation (site), use:
    - `mvn site`
7. **Package the Project**:
    - To create a JAR or WAR file without running tests, use:
    - `mvn package`
8. **Install Dependencies Locally**:
    - To install dependencies into your local repository, use:
    - `mvn install`

How will you configure Jenkins job?

Certainly! To configure a Jenkins job, follow these steps:

1. **Log in to Jenkins**:
    - Access the Jenkins dashboard using your web browser.
2. **Select the Job**:
    - Click on the specific job or item you want to configure.
3. **Navigate to Configuration**:
    - Click on the "Configure" option (usually available in the dropdown menu or on the job page).
    - This will take you to the job configuration page.
4. **Job Configuration**:
    - Modify the job settings as needed. Here are some common configurations:
        - **Description**: Set a brief description for the job.
        - **Build Triggers**: Configure when the job should run (e.g., periodically, on SCM changes).
        - **Source Code Management (SCM)**: Specify the repository URL, credentials, and branch.
        - **Build Environment**: Set environment variables or other build-related settings.
        - **Build Steps**: Define the build steps (e.g., shell commands, Maven goals).
        - **Post-build Actions**: Specify actions to perform after the build (e.g., notifications, archiving artifacts).
5. **Build Triggers**:

- To schedule the job, check the “Build periodically” option.
- Set the desired cron expression (e.g., `H/5 * * * *` for every 5 minutes).

6. **Save Changes**:
   - Click the “Save” button to apply your configuration.

What are two components Jenkins is integrated with?

Jenkins is often integrated with two components: version control systems (VCS) and build tools:

- Version control systems

  Jenkins integrates with VCS like Git, SVN (Apache Subversion), and Mercurial. For example, the Git plugin allows you to clone Git repositories, manage branches, and trigger builds based on Git events.

- Build tools

  Jenkins supports build automation using build tools like Maven, Gradle, and Ant

What is the purpose of version control tool?

Version control tools track changes to files, code, and directories over time, allowing developers to collaborate, compare versions, and revert to previous states. This creates a historical record of how the code evolved, which can help developers fix mistakes and minimize disruption to the team.

What is difference between group id and artifact id?

groupId – a unique base name of the company or group that created the project. artifactId – a unique name of the project. version – a version of the project. packaging – a packaging method (e.g. WAR/JAR/ZIP)