Experiment No. - 6

Student Name: Vivek Kumar
Branch: BE-CSE(LEET)
Semester: 6th
Subject Name: Software Testing Lab

UID: 21BCS8129
Section/Group:20BCS-ST-801/B
Date of Performance: 21/04/2023
Subject Code: 20CSP-380

- **Aim/Overview of the practical:**
  Demonstrate Rest API testing POSTMAN.
- **Theory: Postman** is a scalable API testing tool that quickly integrates into CI/CD pipeline. It started in 2012 as a side project by Abhinav Asthana to simplify API workflow in testing and development. API stands for Application Programming Interface which allows software applications to communicate with each other via API calls.
- **Requirement Analysis: -** Google Chrome, POSTMAN, Online tool.
- **Hardware Requirement: -** Computer, Windows Power Supply.

Why Use Postman?
With over 4 million users nowadays, Postman Software has become a tool of choice for the following reasons:

1. Accessibility - To use Postman tool, one would just need to log-in to their own accounts making it easy to access files anytime, anywhere as long as a Postman application is installed on the computer.

2. Use of Collections - Postman lets users create collections for their Postman API calls. Each collection can create subfolders and multiple requests. This helps in organizing your test suites.

3. 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.

4. Creating Environments - Having multiple environments aids in less repetition of tests as one can use the same collection but for a different environment. This is where parameterization will take place which we will discuss in further lessons.

5. Creation of Tests - Test checkpoints such as verifying for successful HTTP response status can be added to each Postman API calls which help ensure test coverage.

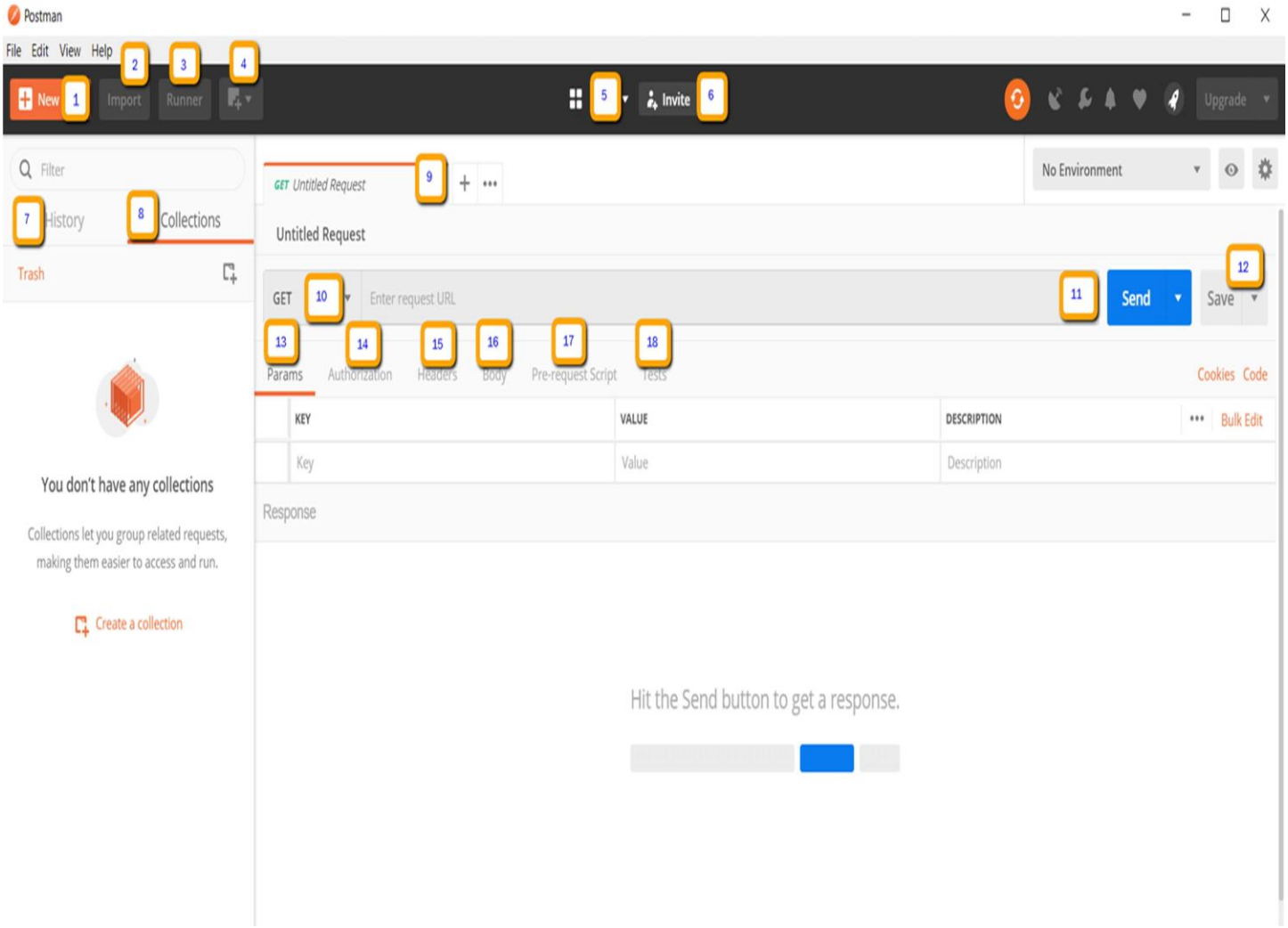
6. Automation Testing - Through the use of the Collection Runner or Newman, tests can be run in multiple iterations saving time for repetitive tests.

7. Debugging - Postman console helps to check what data has been retrieved making it easy to debug tests.

8. Continuous Integration - With its ability to support continuous integration, development practices are maintained.

How to use Postman to execute APIs

Below is the Postman Workspace. Let's explore the step-by-step process on How to use Postman and different features of the Postman tool!



1. New-This is where you will 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 raw text.

3. Runner - Automation tests can be executed through the Collection Runner. This will be discussed further in the next lesson.

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 - Collaborate on a workspace by inviting team members.

7. History - Past requests that you have sent will be displayed in History. This makes it easy to track actions that you have done.

8. Collections - Organize your test suite by creating collections. Each collection may have subfolders and multiple requests. A request or folder can also be duplicated as well.

9. Request tab - This displays the title of the request you are working on. By default, "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 Postman API testing, the most commonly used requests are GET and POST.

11. Request URL -Also known as an endpoint, this is where you will identify the link to where the API will communicate with.

12. Save -If there are changes to a request, clicking save is a must so that new changes will not be lost or overwritten.

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

14. Authorization -In order to access APis, proper authorization is needed. It may be in the form of a usemame 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 request.

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 tests as it sets up checkpoints to verify if response status is ok, retrieved data is as expected and other tests.

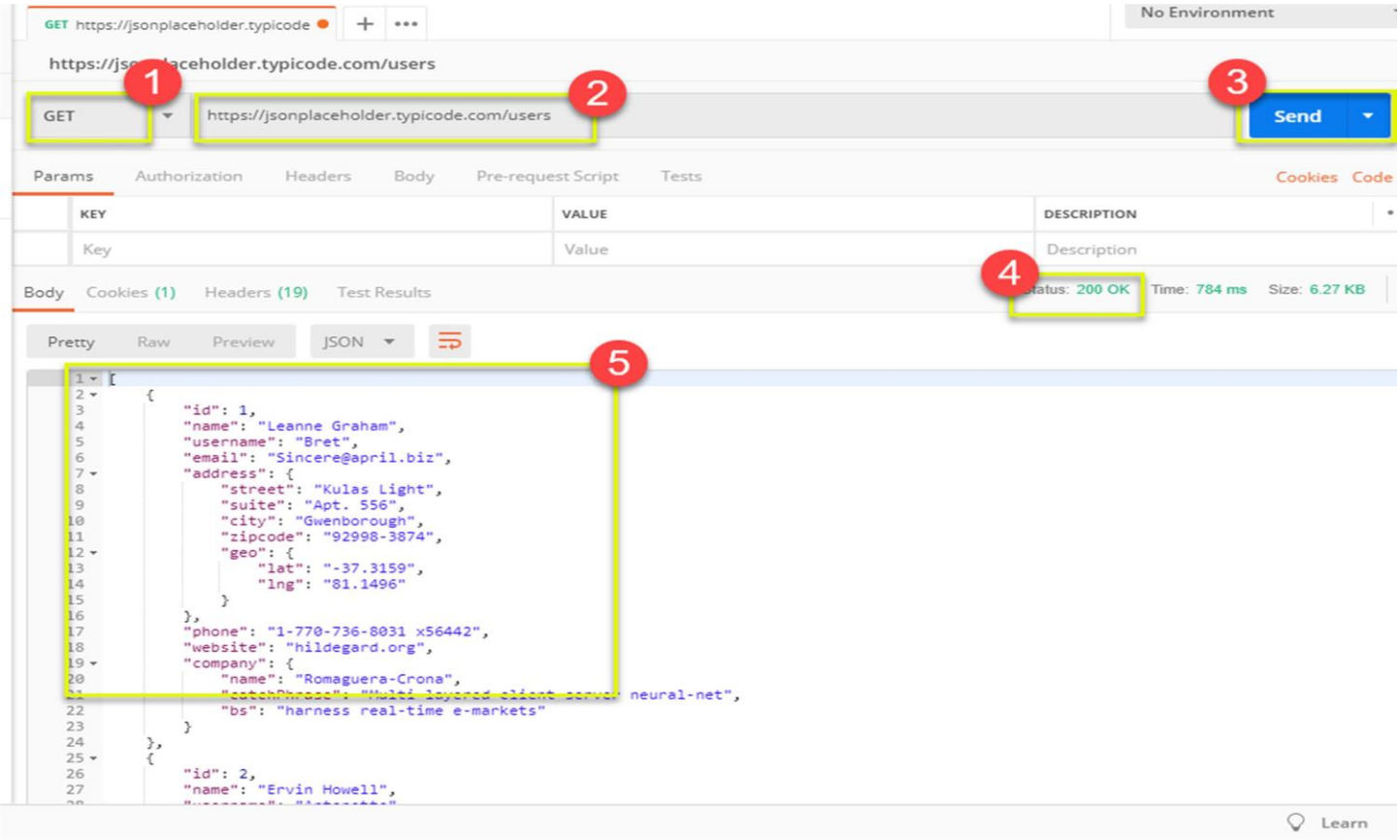
Working with GET Requests:

Get requests are used to retrieve information from the given URL. There will be no changes done to the endpoint. We will use the following URL for all examples in this Postman tutorial.

https://j sonplaceholder .typicode.com/users

In the workspace

1. Set your HTTP request to GET.
2. In the request URL field, input link
3. Click Send
4. You will see 200 OK Message
5. There should be 10 user results in the body which indicates that your test has run successfully.

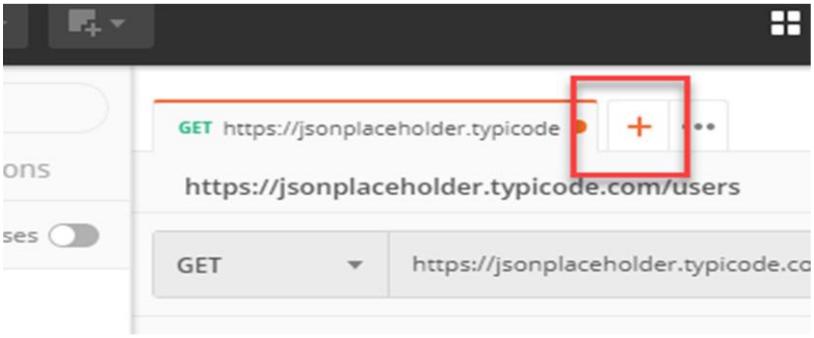


***Note:** There may be cases that Get Postman request may be unsuccessful.

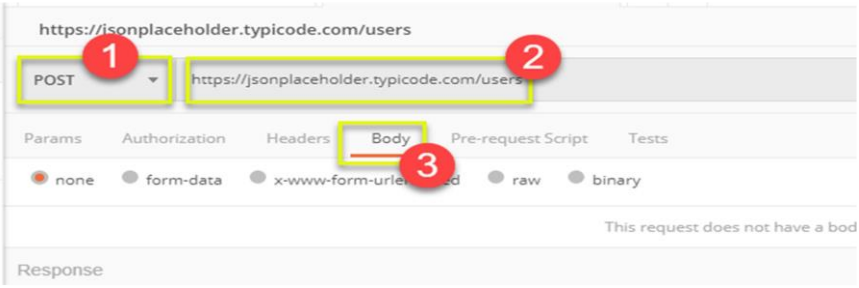
Working with POST Requests

Post requests are different from Get request as there is data manipulation with the user adding data to the endpoint. Using the same data from the previous tutorial in Get request, let's now add our own user.

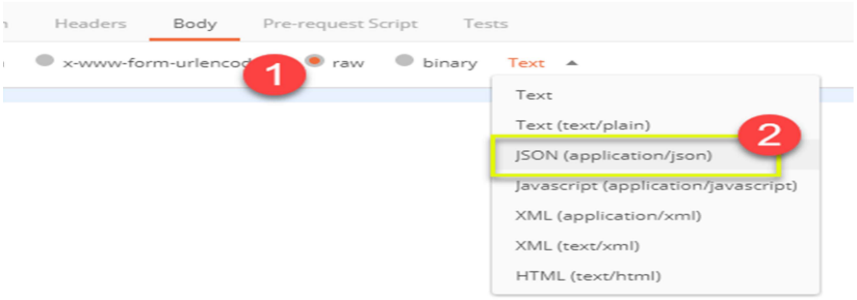
Step 1) Click a new tab to create a new request.



Step 2) In the new tab
Set your HTTP request to POST.
Input the same link in request url: https://jsonplaceholder.typicode.com/users
Switch to the Body tab

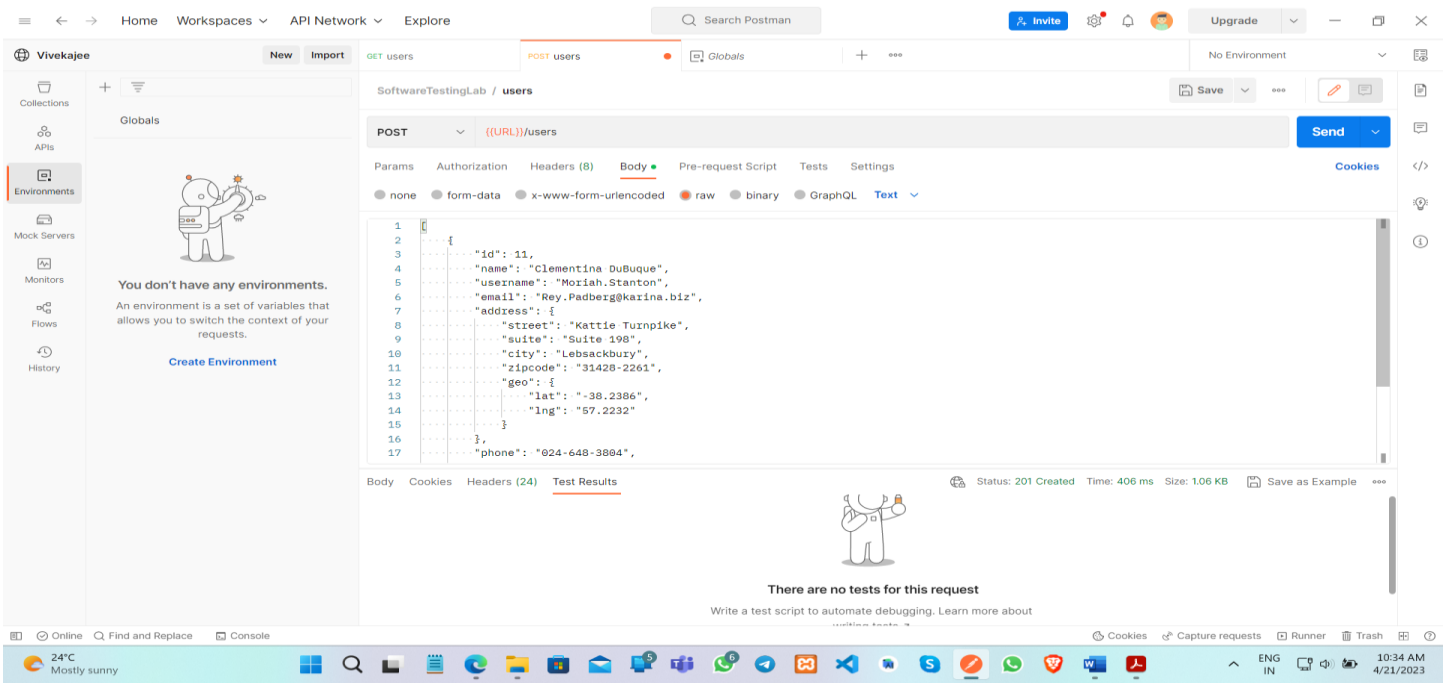


Step 3) In Body,
Click Raw
Select JSON.



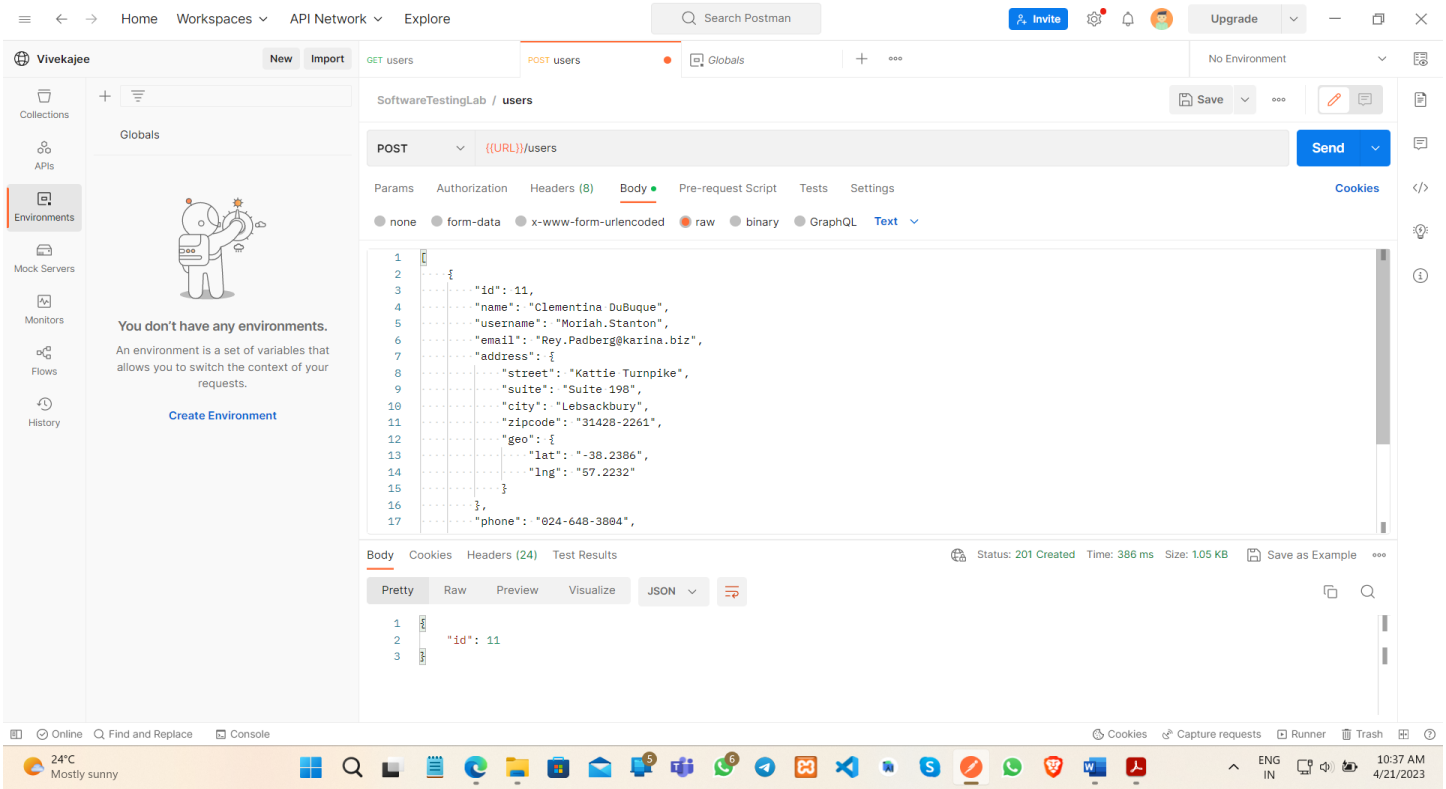
Step 4) Copy and paste just one user result from the previous get request like below. Ensure that the code has been copied correctly with paired curly braces and brackets. Change id to 11 and name to any desired name. You can also change other details like the address.


```
[
  {
    "id": 11,
    "name": "Clementina DuBuque",
    "username": "Moriah.Stanton",
    "email": "Rey.Padberg@karina.biz",
    "address": {
      "street": "Kattie Turnpike",
      "suite": "Suite 198",
      "city": "Lebsackbury",
      "zipcode": "31428-2261",
      "geo": {
        "lat": "-38.2386",
        "lng": "57.2232"
      }
    },
    "phone": "024-648-3804",
    "website": "ambrose.net",
    "company": {
      "name": "Hoeger LLC",
      "catchPhrase": "Centralized empowering task-force",
      "bs": "target end-to-end models"
    }
  }
]
```



***Note:** Online Post request should have the correct format to ensure that requested data will be created. It is a good practice to use Get first to check the JSON format of the request. You can use tools like https://jsonformatter.curiousconcept.com/

Step 5) Next,
Click Send.
Status: 201 Created should be displayed
Posted data are showing up in the body.



Learning outcomes (What I have learnt):

We have learned the Use of POSTMAN

We have learned the Get and Post Method uses.

Evaluation Grid (To be created per the faculty's SOP and Assessment guidelines):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet completion including writing learning objectives/Outcomes. (To be submitted at the end of the day).		
2.	Post-Lab Quiz Result.		
3.	Student Engagement in Simulation/Demonstration/Performance and Controls/Pre-Lab Questions.		
	Signature of Faculty (with Date):	Total Marks Obtained:	