How to use Postman tool for API testing?

* **Working with GET Requests**

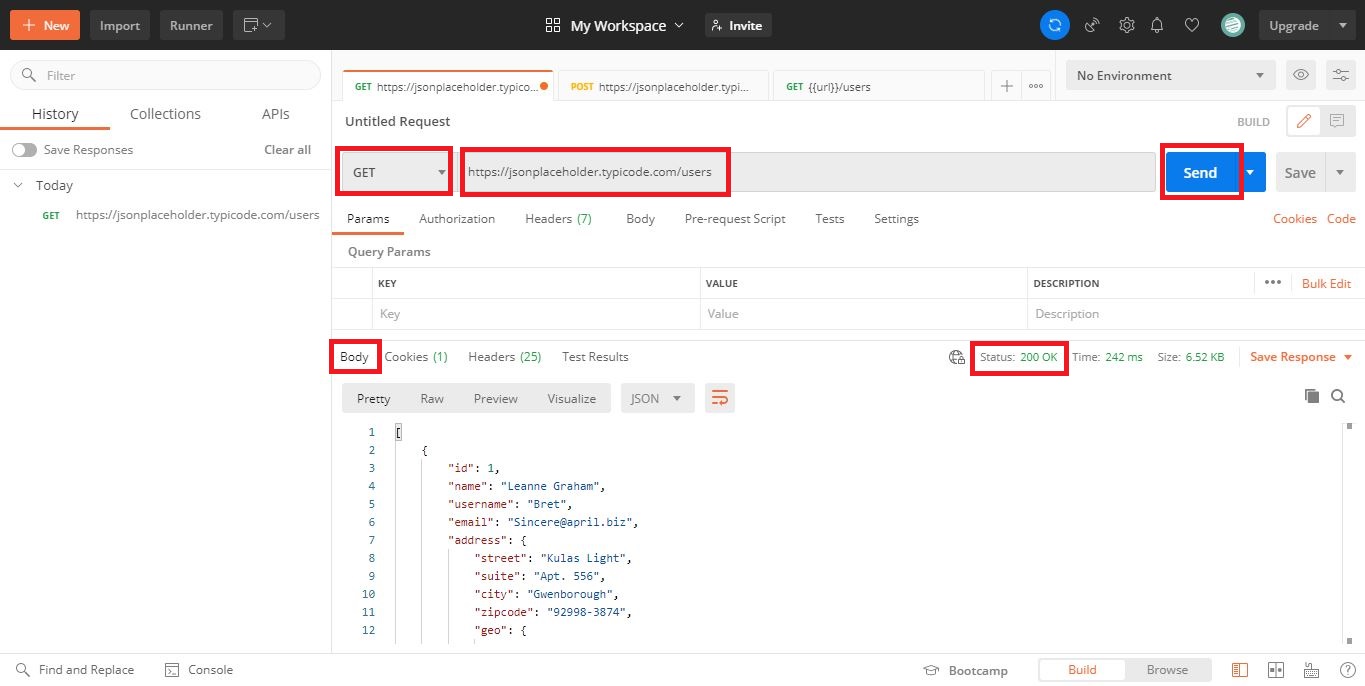
**Get requests are used to retrieve information from the given URL**. There will be no changes done to the endpoint.

The following URL will be used for all examples in this tutorial:

<https://jsonplaceholder.typicode.com/users>

In the workspace:

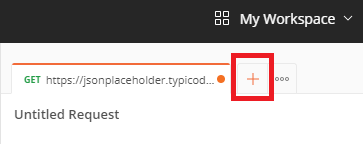
1. Set your **HTTP request** to **GET**.
2. In the request **URL field**, input link.
3. Click **Send.**
4. You will see ***200 OK* Message**.
5. There should be 10 user results in the body which indicates that your test has run successfully.



* **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.

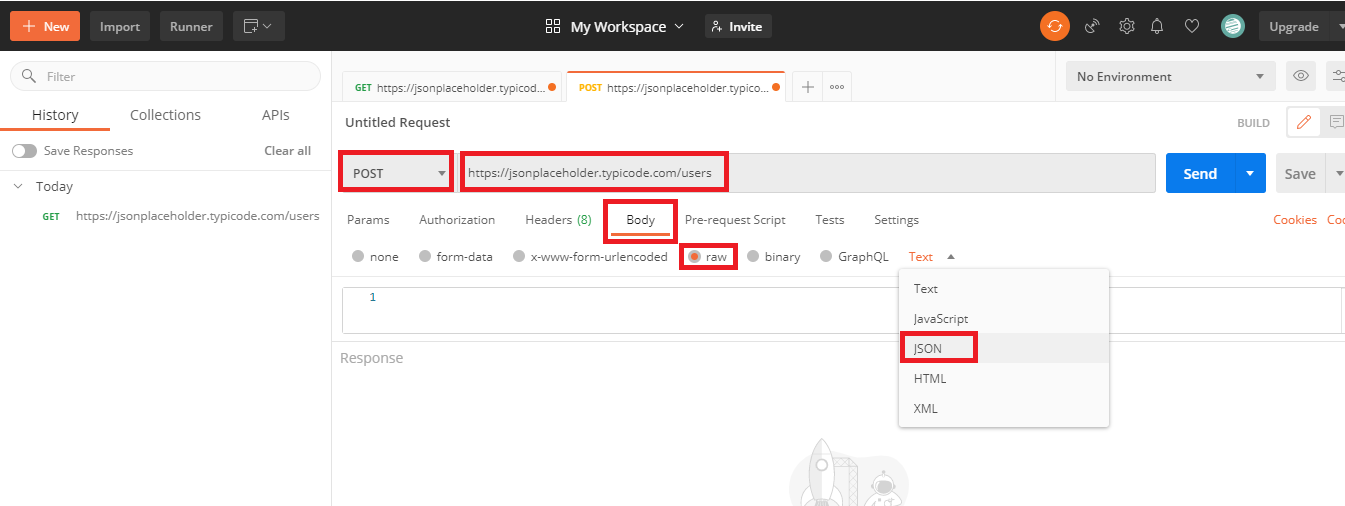


**Step 2)**  In the new tab:

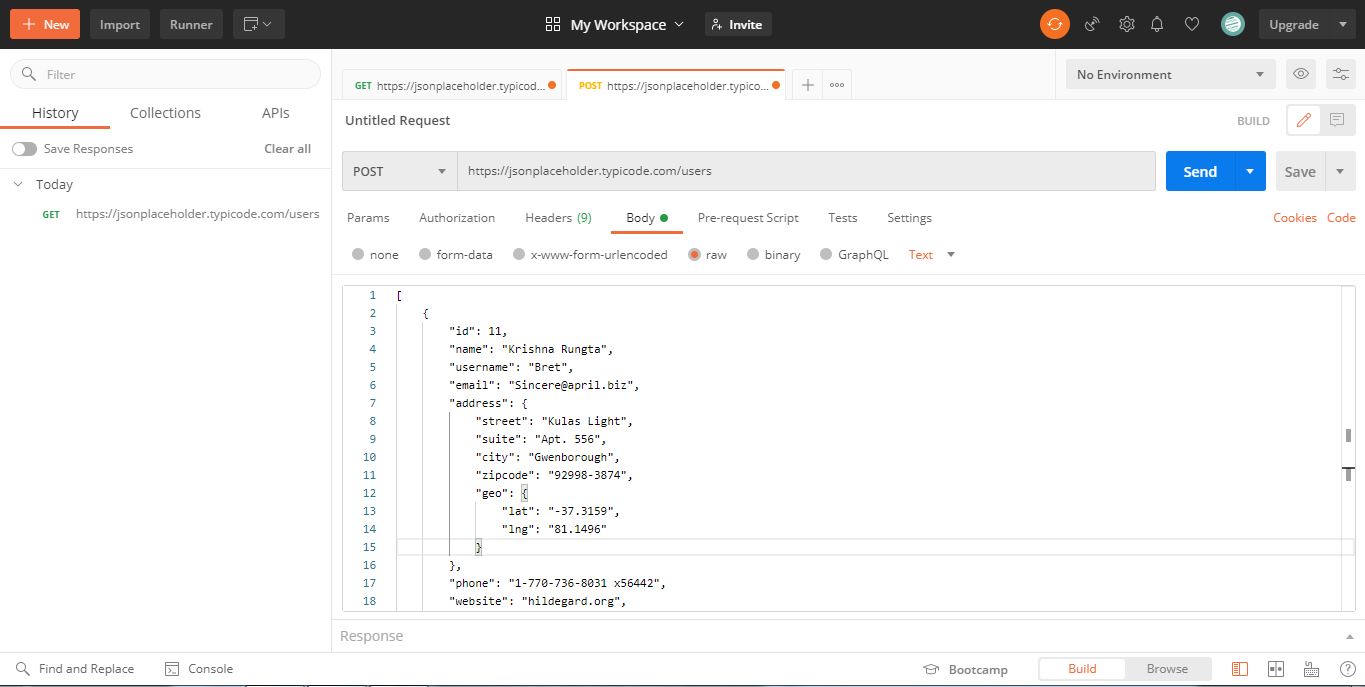
1. Set your **HTTP request** to **POST**.
2. Input the same link in **request url**:

<https://jsonplaceholder.typicode.com/users>

1. In **Body**, click **Row** and Select **JSON**

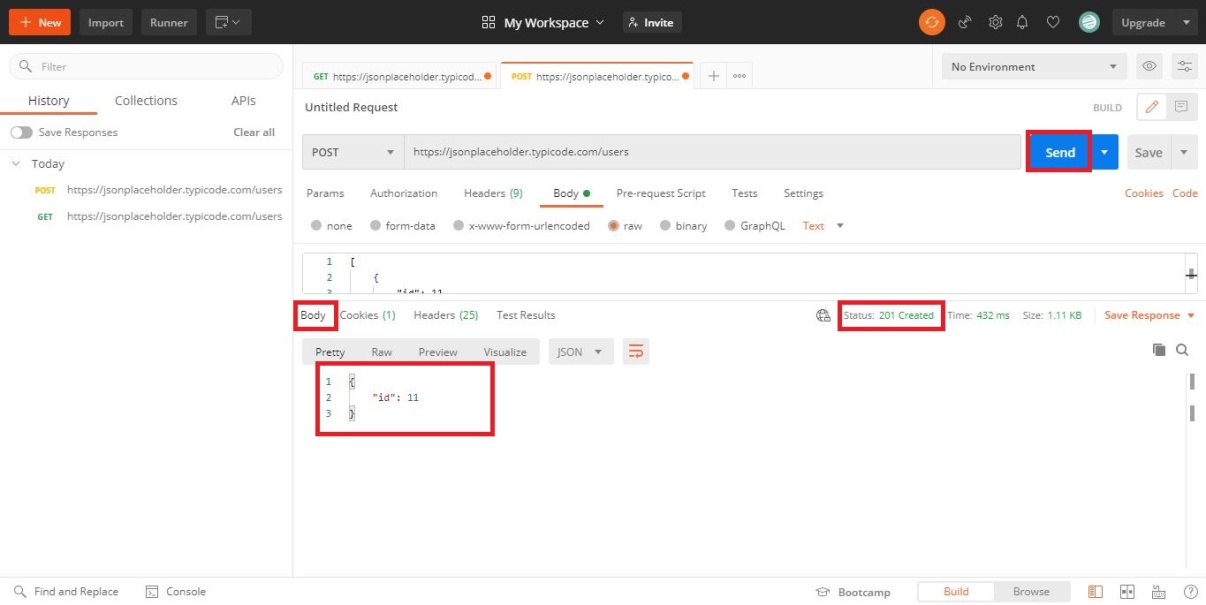


**Step 3)** Copy and paste just one user result from the previous get request like below. Ensure that the code has been copied correctly with paired curly braces and brackets. Change id to 11 and name to any desired name. You can also change other details like the address.



**Step 4)** Next,

1. Click**Send**.
2. **Status: 201 Created** should be displayed.
3. Posted data are showing up in the body.



* **How to Parameterize Requests?**

**Data Parameterization**is one of the most useful features of **Postman**. Instead of creating the same requests with different data, you can use variables with parameters. These data can be from a data file or an environment variable. **Parameterization** helps to avoid repetition of the same tests and iterations can be used for automation testing.

Parameters are created through the use of double curly brackets: {{sample}}.

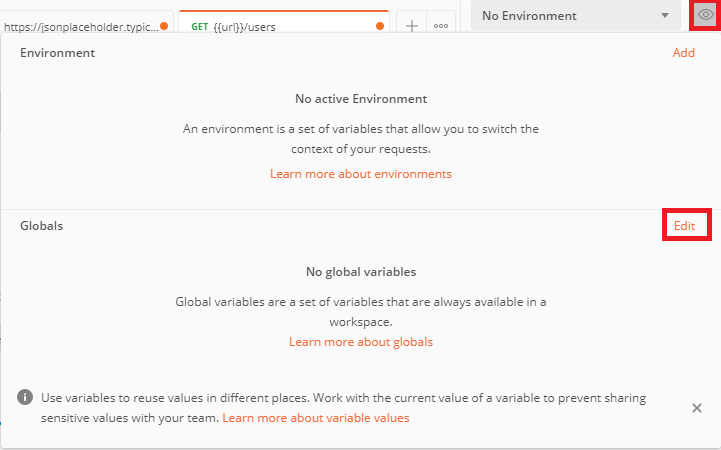
Now let's create a parameterize get request.

**Step 1)**

1. Set your **HTTP request** to **GET**
2. Input this link: <https://jsonplaceholder.typicode.com/users>. Replace the first part of the link with a parameter such as {{url}}. Request url should now be {{url}}/users.
3. Click **send**.

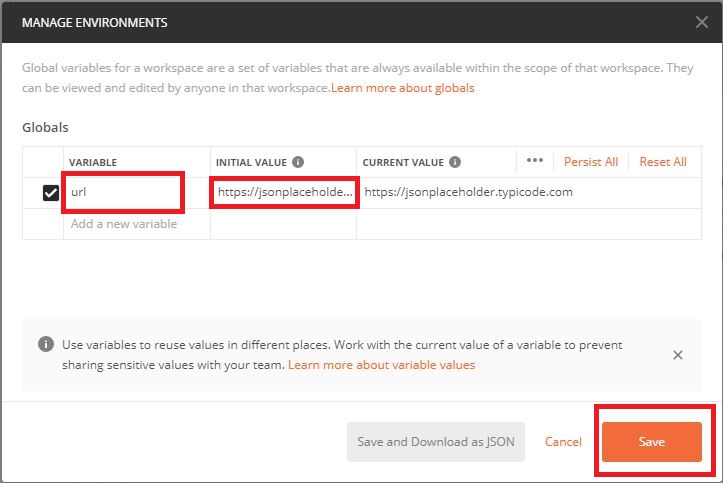
**Step 2)**To use the parameter you need to set the environment:

1. Click the **eye** icon.
2. Click **edit** to set the variable to a global environment which can be used in all collections.

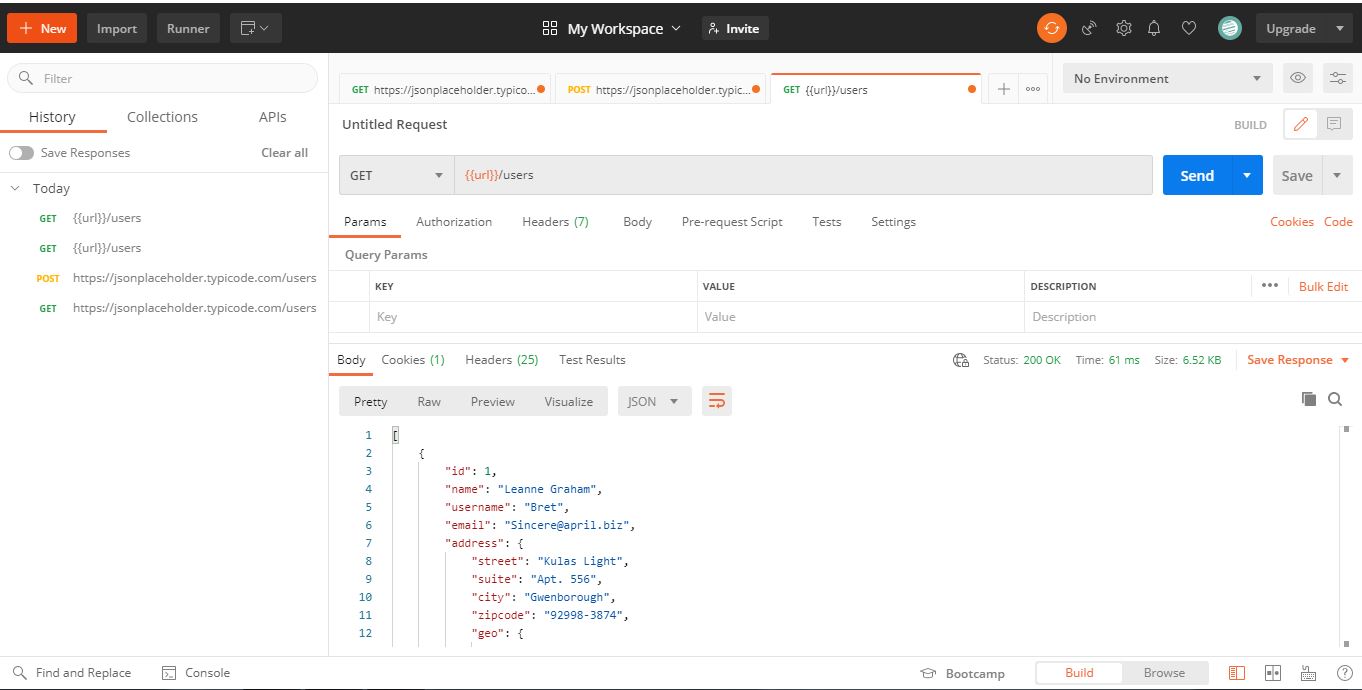


**Step 3)** In variable,

1. set the name to the **url** which is [https://jsonplaceholder.typicode.com](https://jsonplaceholder.typicode.com/)
2. click**Save**.



**Step 5)**Go back to your **Get request** then click **Send**. There should now be results for your request.

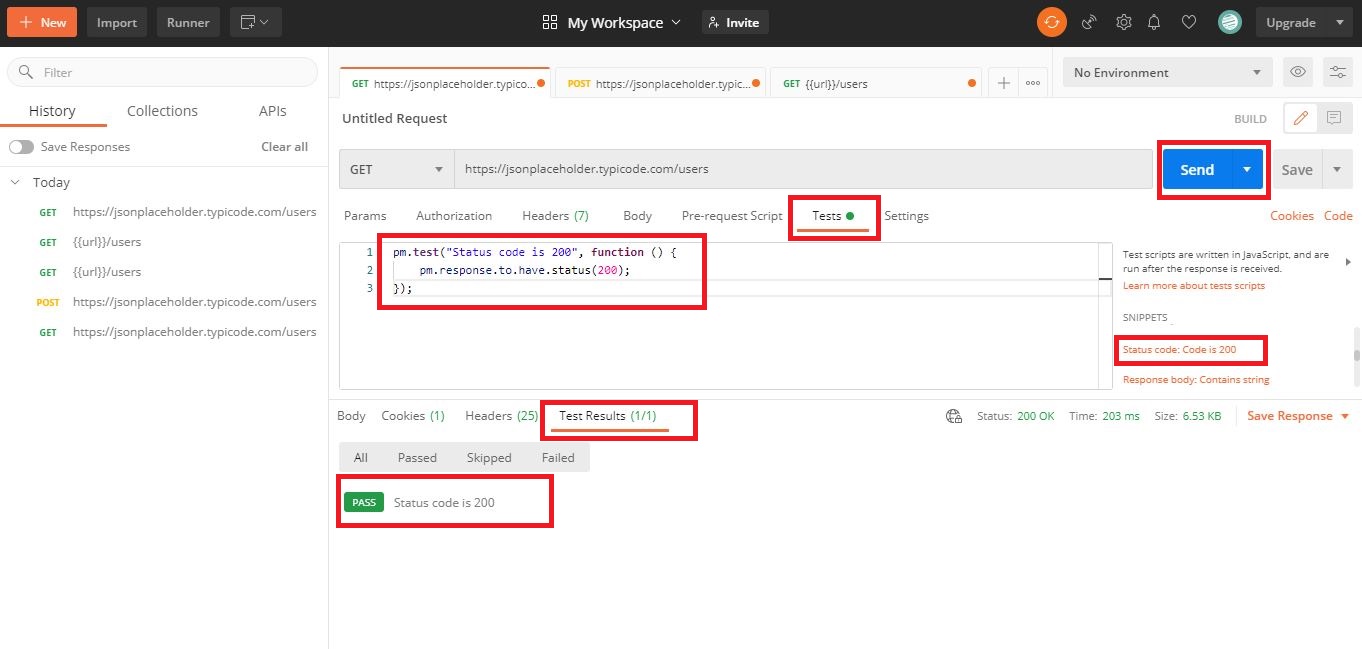


* **How to Create Postman Tests?**

**Postman Tests are JavaScript codes added to requests that help you verify results such as successful or failed status, comparison of expected results,**etc**.** It usually starts with pm.test. It can be compared to asserts, verify commands available in other tools.

**Step 1)**Go to your **GET user request** from the previous tutorial.

1. Switch to the **tests tab**. On the right side are snippet codes.
2. From the snippets section, click on **"Status code: Code is 200"**.
3. Click **Send**. The **test result** should now be displayed.



**Step 3)**Go back to the test tab and let's add another test. This time will be compared the expected result to the actual result.

From the **snippets section**, click on "**Response body:JSON value check**". This way it will be checked if Leanne Graham has the userid 1.

**Step 4)**

1. Replace "Your Test Name" from the code with "Check if user with id1 is Leanne Graham" so that the test name specifies exactly what we want to test.
2. Replace jsonData.value with jsonData[0].name. To get the path, check the body in **Get result** earlier. Since Leanne Graham is userid 1, jsonData is in the first result which should start with 0. If you want to get the second result, use jsonData[1] and so on for succeeding results.
3. In to eql, input "Leanne Graham".

