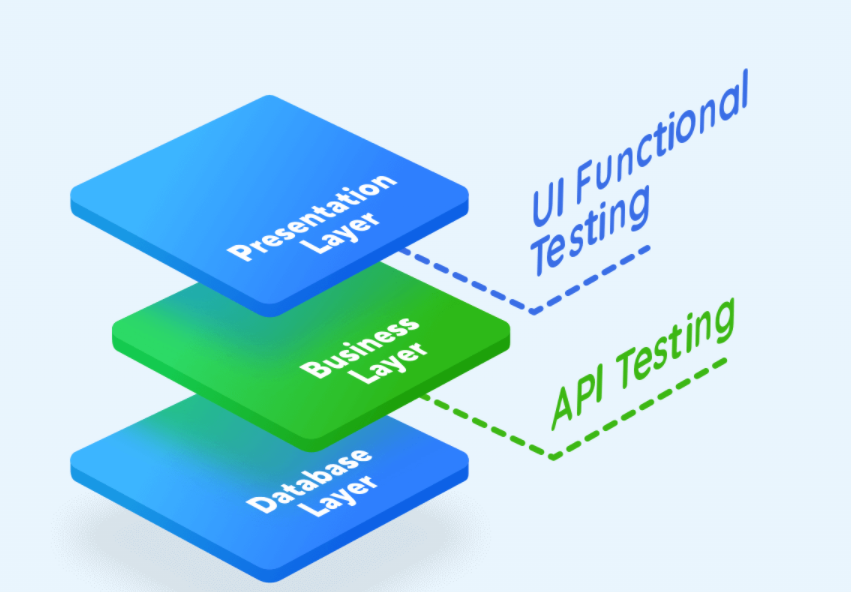
What is API testing?

API stands for Application Programming Interface.

In software application (app) development, API is the middle layer between the presentation (UI) and the database layer. APIs enable communication and data exchange from one software system to another.

API testing is a software testing practice that tests the APIs directly — from their functionality & reliability. Part of integration testing, API testing effectively validates the logic of the build architecture within a short amount of time.



**Benefits of API testing**

**Language-independent**

Data is exchanged via XML and JSON formats, so any language can be used for test automation. XML and JSON are typically structured data, making the verification fast and stable. There are also built-in libraries to support comparing data using these data formats.

**GUI-independent**

API testing can be performed in the app prior to GUI testing. Early testing means early feedback and better team productivity. The app's core functionalities can be tested to expose small errors and to evaluate the build's strengths.

**Improved test coverage**

Most API/web services have specifications, allowing you to create automated tests with high coverage — including functional testing and non-functional testing.

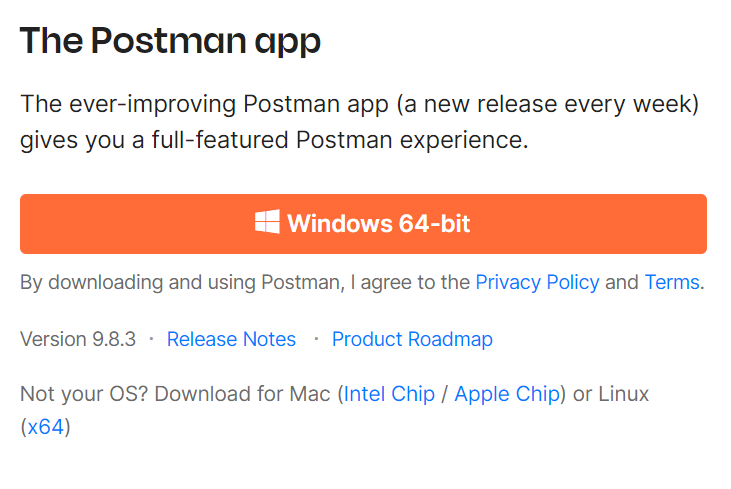
**Faster releases**

It is common that executing API testing saves up to eight hours compared to UI testing, allowing software development teams to release products faster.

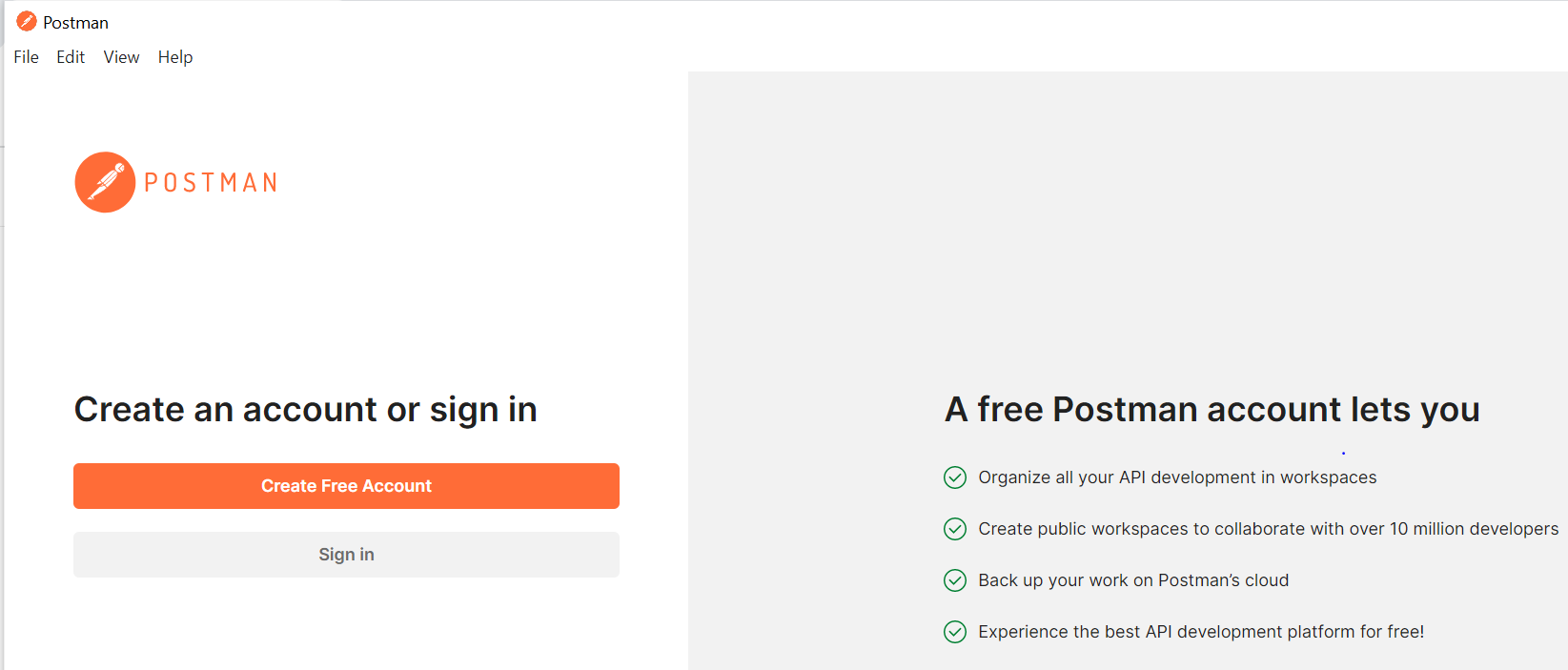
## Download and Install POSTMAN

Being an Open Source tool, Postman can be easily downloaded. Here are the steps to install:

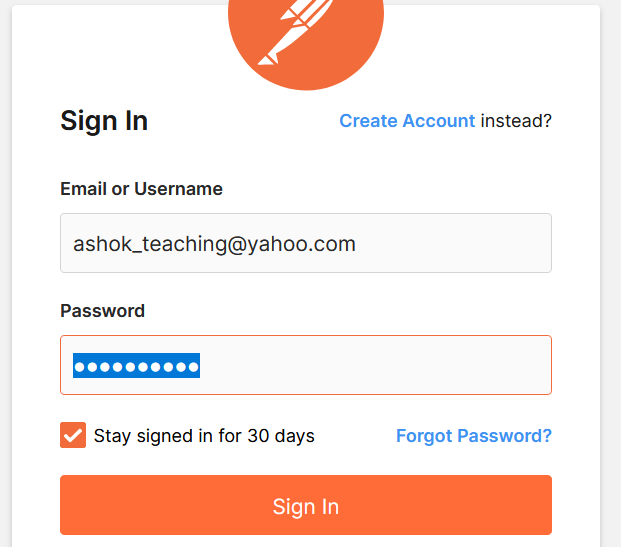
1. Go to <https://www.postman.com/downloads/> and choose your desired platform among Mac, Windows or Linux. Click Download.



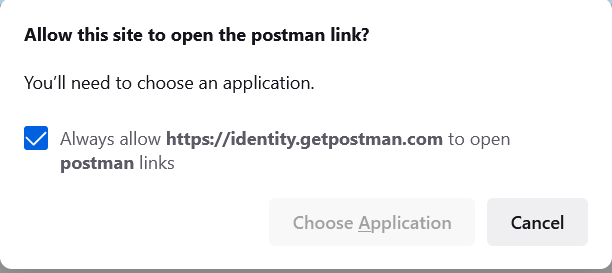
1. Once you run it will open postman as shown below



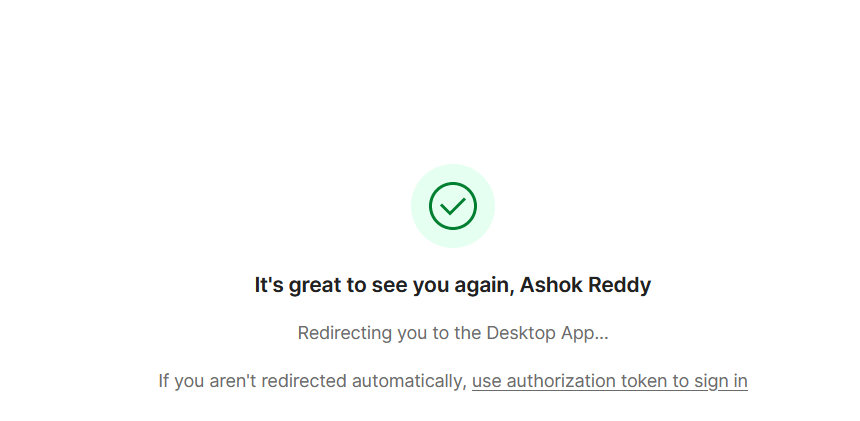
1. Click on Sign in Account
2. Enter email id and password



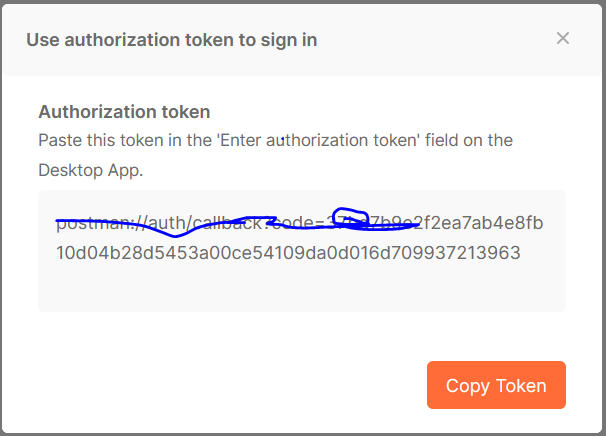
1. Click on Choose Application



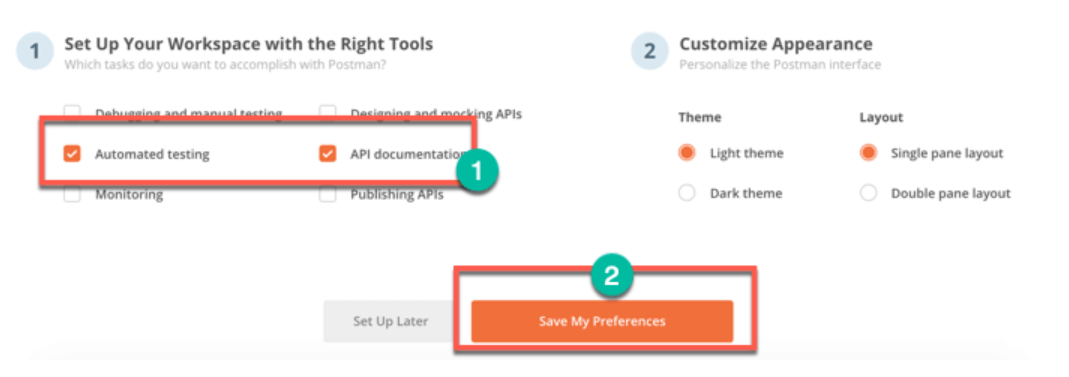
1. Wait for it to open



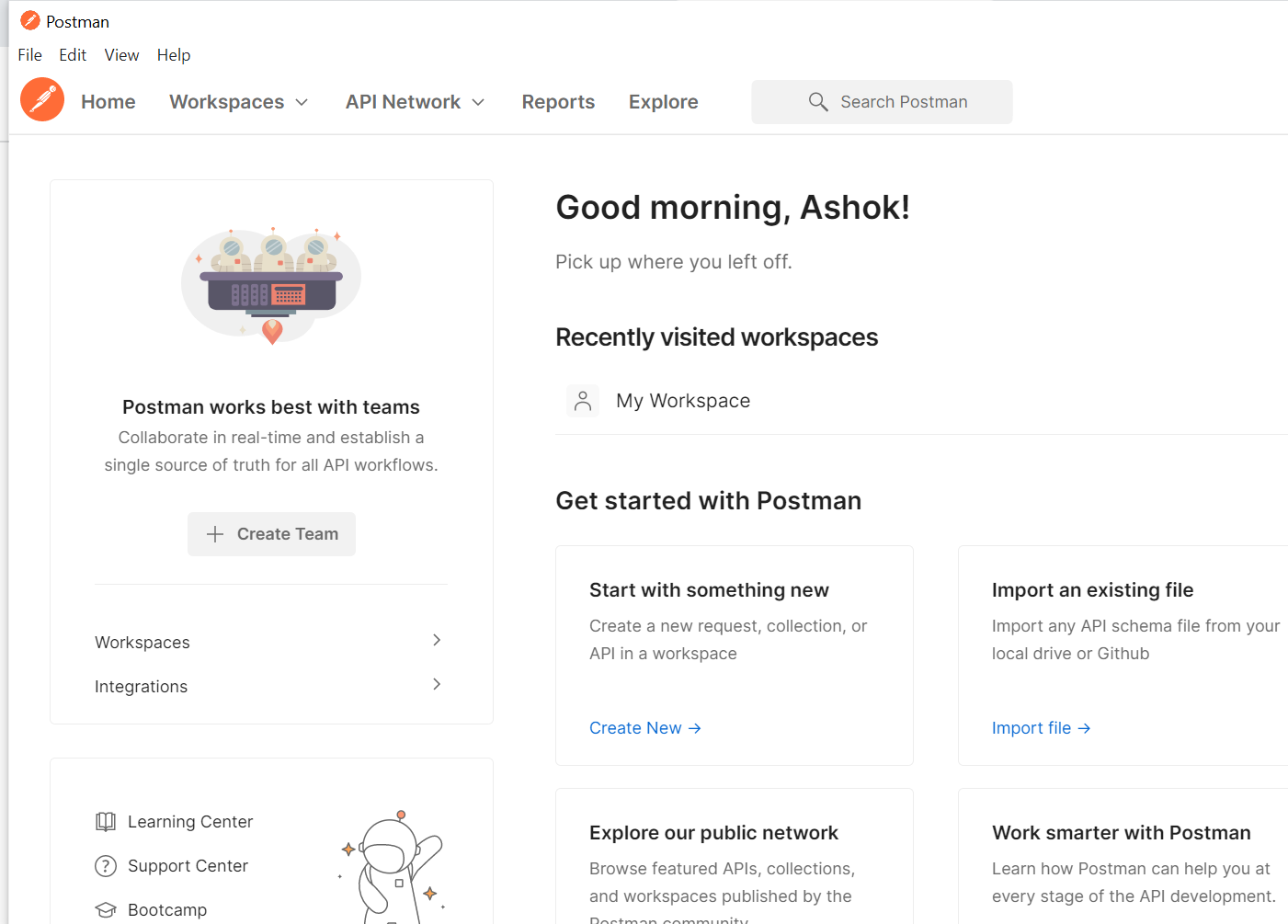
1. If it is not opening click on use authorization token to sign in



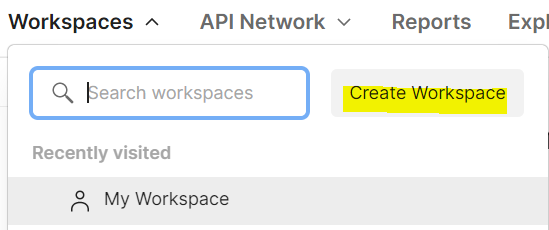
1. Once opened setup your preferences as shown below



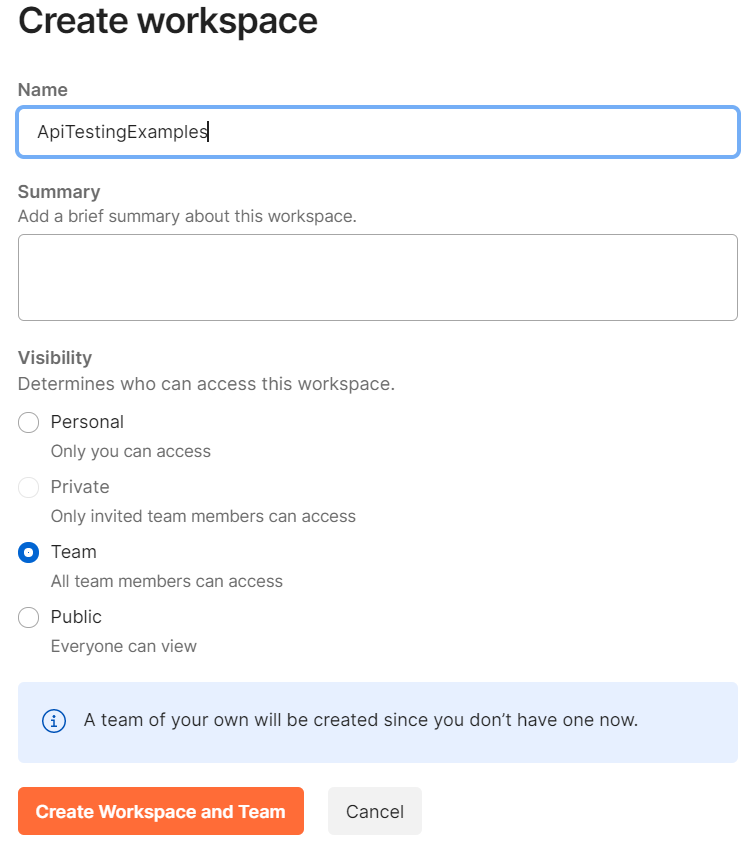
1. Once saved preferences you will see below screen and it is ready to use



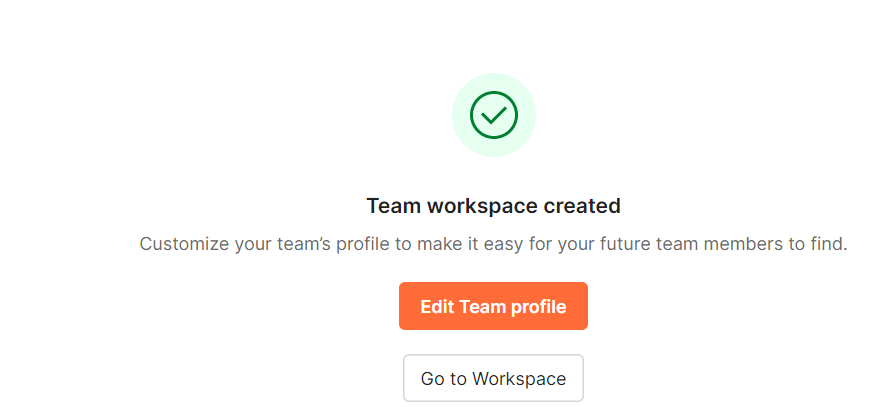
1. From workspaces select as shown below

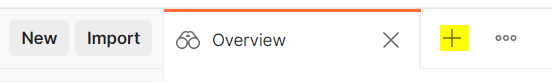


1. Provide name and other information as per the below and click on create workspace and team



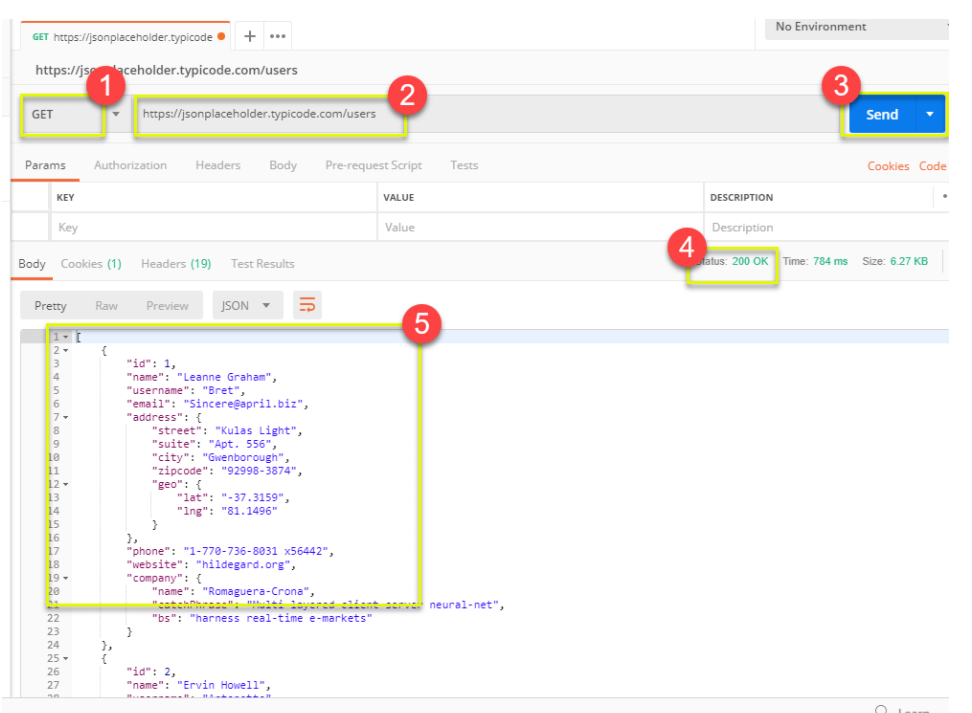
Click on Go to Workspace<https://jsonplaceholder.typicode.com/users>

1. 
2. In workspace click on + icon as show below

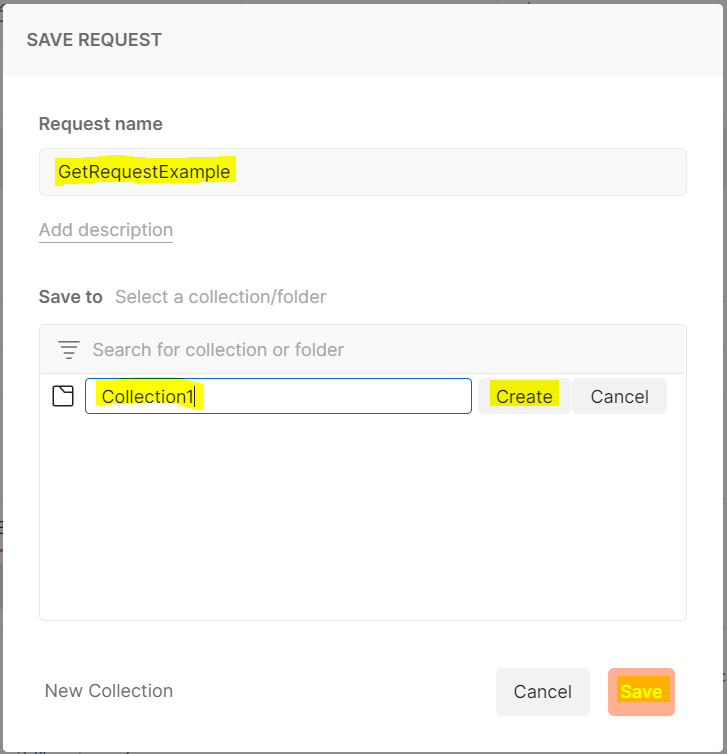


**Working with Get Request:**

1. In the workspace follow below steps as shown
   1. Set your HTTP request to GET.
   2. In the request URL field enter URL <https://jsonplaceholder.typicode.com/users> , 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.



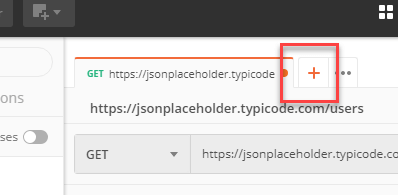
1. Follow below steps to save your request



## Working with POST Requests

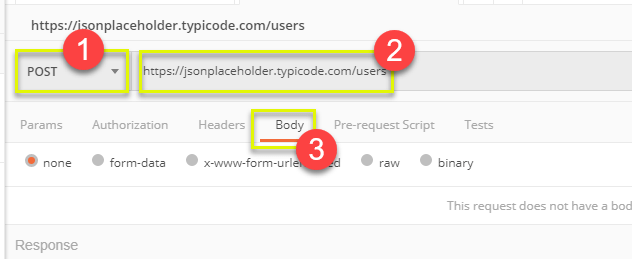
Post requests are different from Get request as there is data manipulation with the user adding data to the endpoint. Using the same data from the previous tutorial in Get request, let’s add one new user.

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



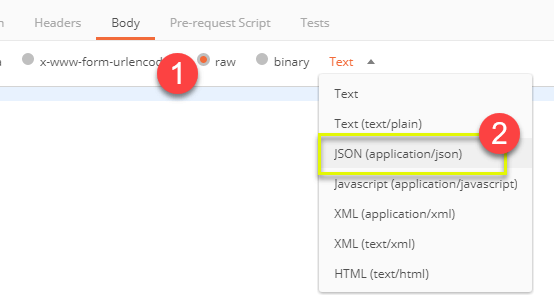
**Step 2)** In the new tab

1. Set your HTTP request to POST.
2. Input the same link in request url: <https://jsonplaceholder.typicode.com/users>
3. switch to the Body tab



**Step 3)** In Body,

1. Click raw
2. 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": "Ashok Reddy",

"username": "Ashok",

"email": "ashok\_testing@yahoo.com",

"address": {

"street": "Kulas Light",

"suite": "Apt. 556",

"city": "Gwenborough",

"zipcode": "92998-3874",

"geo": {

"lat": "-37.3159",

"lng": "81.1496"

}

},

"phone": "1-770-736-8031 x56442",

"website": "hildegard.org",

"company": {

"name": "Romaguera-Crona",

"catchPhrase": "Multi-layered client-server neural-net",

"bs": "harness real-time e-markets"

}

}

]

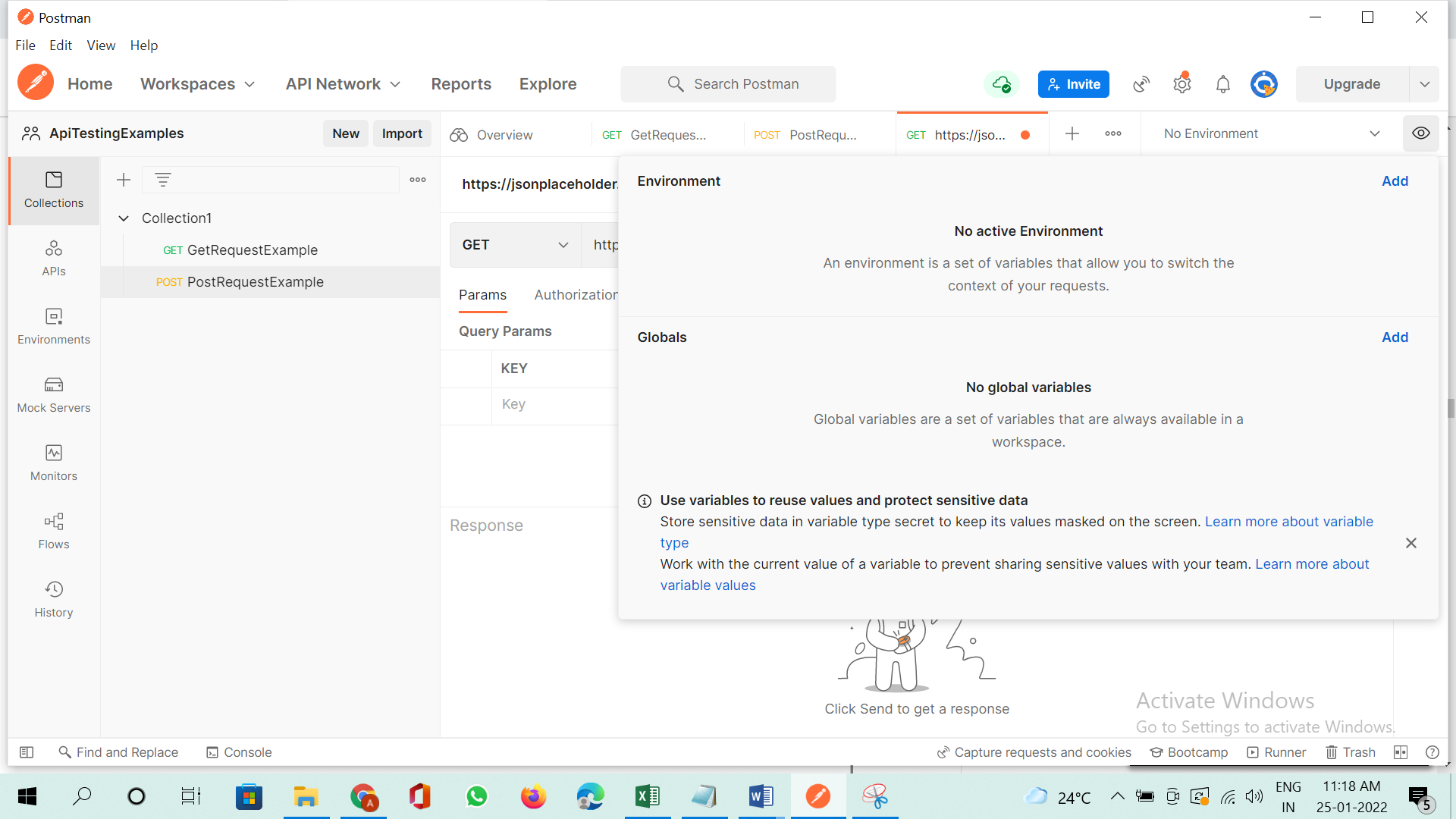
Step 5) We can see new record is added as shown below with status code as 201 Created:



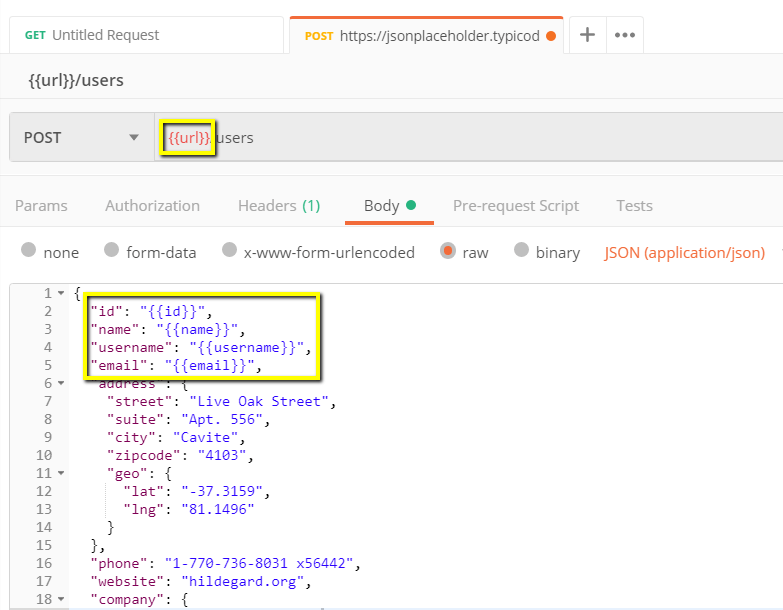
**\*Note:**Online Post request should have the correct format to ensure that requested data will be created. It is a good practice to use Get first to check the JSON format of the request. You can use tools to validate whether given JSON data is right like <https://jsonformatter.curiousconcept.com/>

## 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



Parameters are created through the use of double curly brackets: {{sample}}. Let’s take a look at an example of using parameters in our previous request:

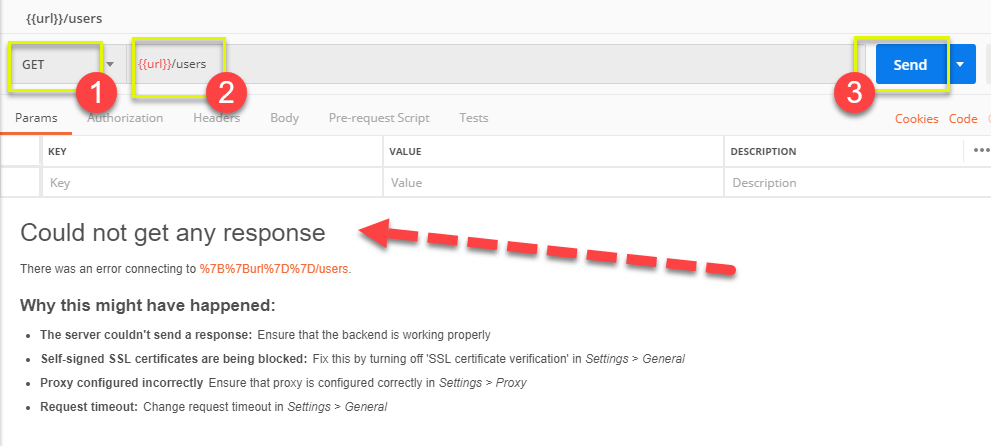


Now let’s create a parameterize get request.

**Step 1)**

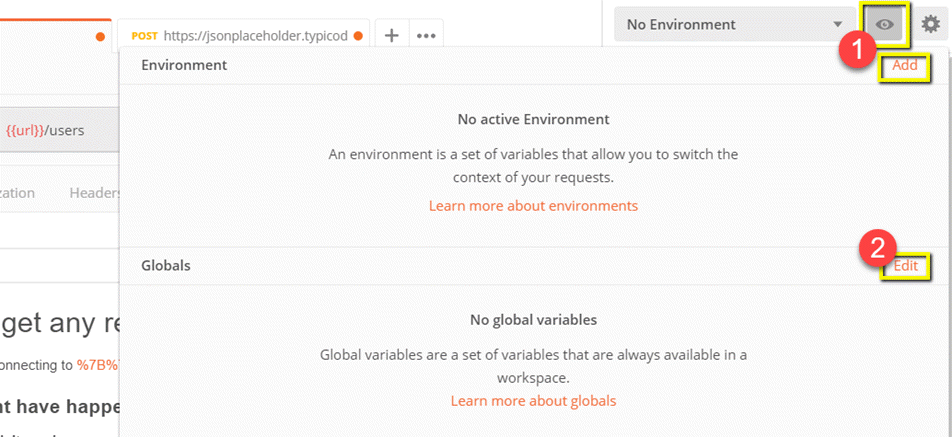
1. Set your HTTP request to GET
2. Input this link: <https://jsonplaceholder.typicode.com/users>. Replace the first part of the link with a parameter such as {{url}}. Request url should now be {{url}}/users.
3. Click send.

There should be no response since we have not set the source of our parameter.



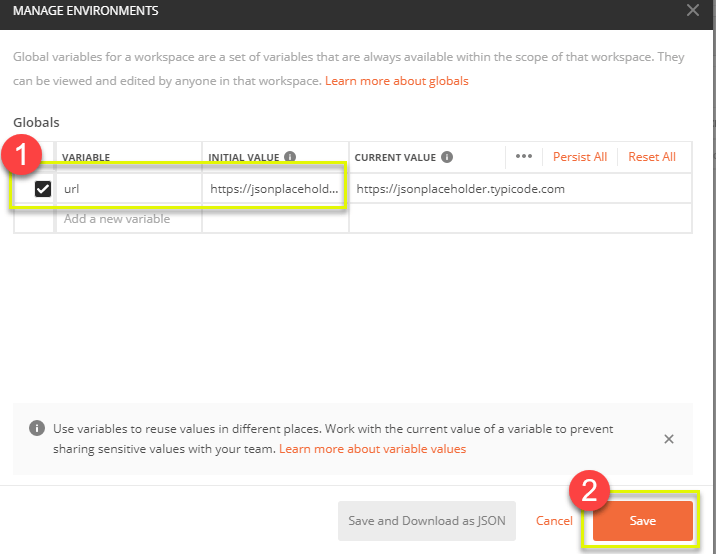
**Step 2)**To use the parameter you need to set the environment

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

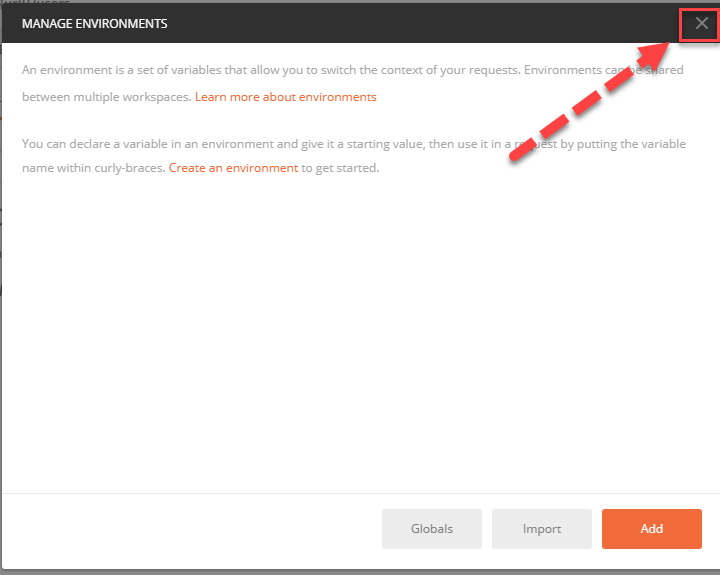


**Step 3)** In variable,

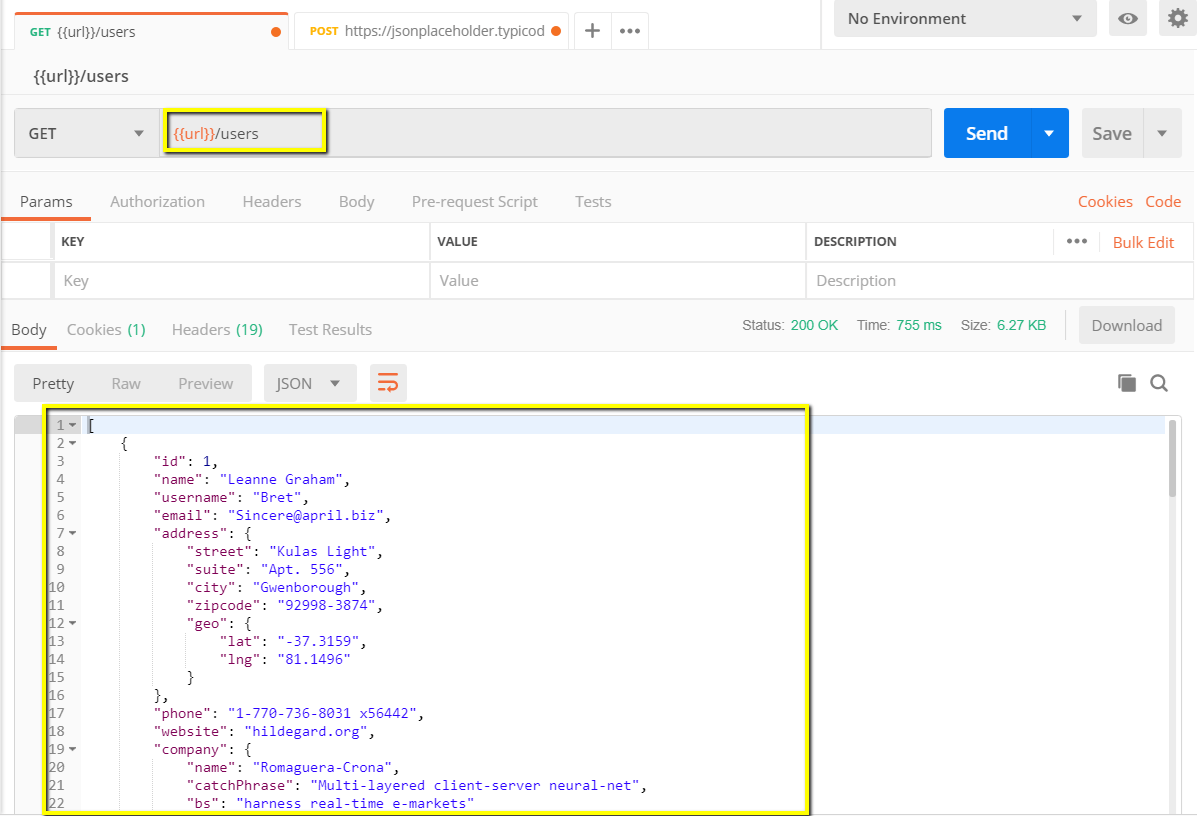
1. set the name to the url which is https://jsonplaceholder.typicode.com
2. click Save.



**Step 4)**Click close if you see the next screen



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



**\*Note:** Always ensure that your parameters have a source such as an environment variable or data file to avoid errors.

## How to Create Postman Tests

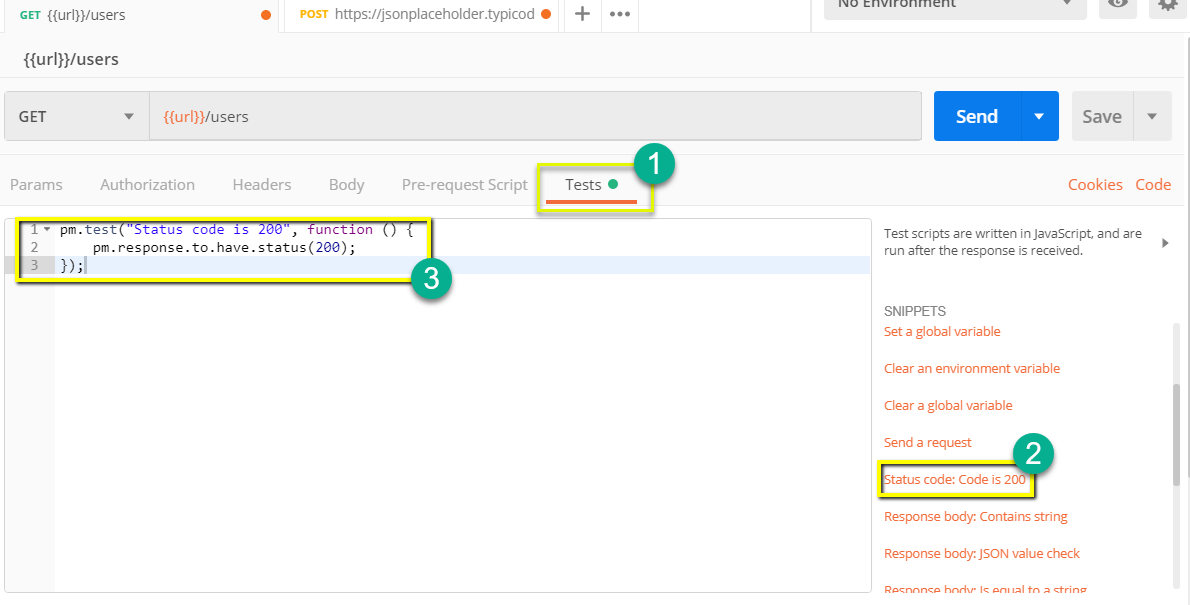
Postman Tests are JavaScript codes added to requests that help you verify results such as successful or failed status, comparison of expected results, etc. It usually starts with pm.test. It can be compared to asserts, verify commands available in other tools.

Let’s do some basic [API testing](https://www.guru99.com/api-testing.html) using Postman for our parameterize requests from the previous lesson.

**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”.

The pane is auto-populated

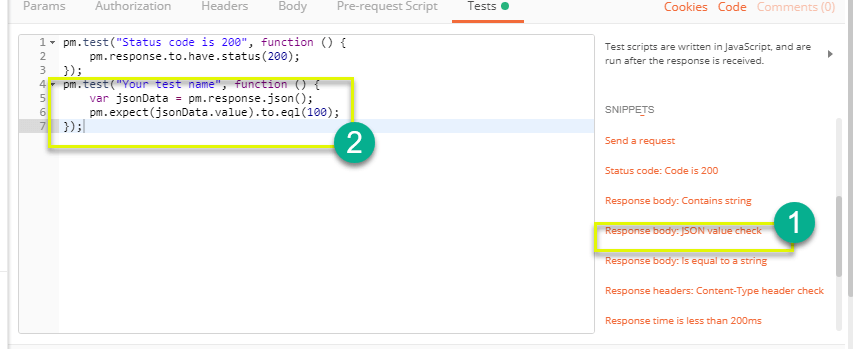


**Step 2)**Now click Send. The test result should now be displayed.



**Step 3)**Go back to the test tab and let’s add another test. This time we will compare the expected result to the actual result.

From the snippets section, click on “Response body:JSON value check”. We will be checking if Leanne Graham has the userid 1.



**Step 4)**

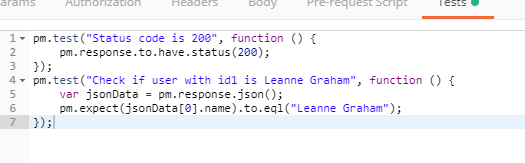
1. Replace “Your Test Name” from the code with “Check if user with id1 is Leanne Graham” so that the test name specifies exactly what we want to test.
2. Replace jsonData.value with jsonData[0].name. To get the path, check the body in Get result earlier. Since Leanne Graham is userid 1, jsonData is in 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”

pm.test("Check if user with id1 is Leanne Graham", function () {

var jsonData = pm.response.json();

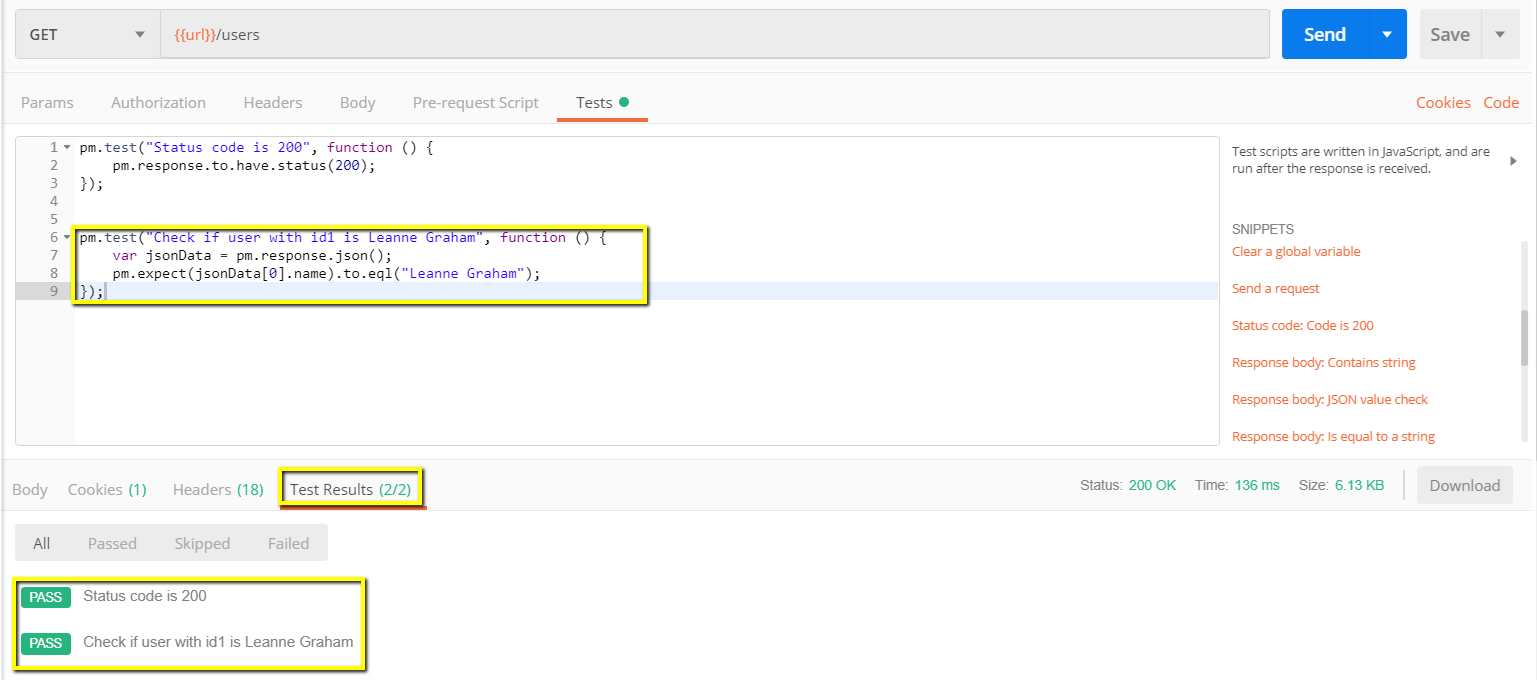
pm.expect(jsonData[0].name).to.eql("Leanne Graham");

});



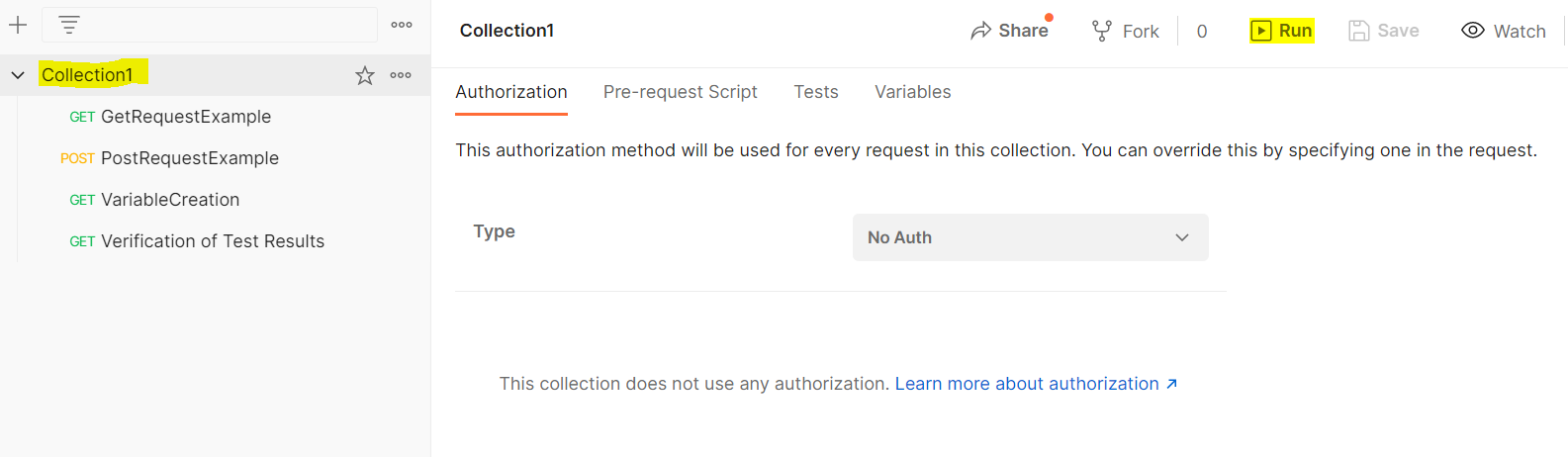
**Step 5)**Click send. There should now be two passed test results for your request.

**\*Note:** There are different kind of tests that can be created in Postman. Try to explore the tool and see what tests will fit your needs.



## How to Run Collections using Collection Runner

* 1. Click on Collections then click on Run as shown below



1. Select tests you want to run and set parameters as shown below

