# POSTMAN

## 1. Introduction to postman

**Postman** is a scalable API testing tool that quickly integrates into CI/CD pipeline.

A continuous integration and continuous deployment (CI/CD) pipeline is **a series of steps that must be performed in order to deliver a new version of software**. CI/CD pipelines are a practice focused on improving software delivery throughout the software development life cycle via automation.

What is API testing?

API testing is **a type of software testing that analyzes an application program interface (API) to verify it fulfills its expected functionality, security, performance and reliability**. The tests are performed either directly on the API or as part of integration testing.

## 2.Download and install postman

1.1 download postman (windows 64)

1.2 Open postman windows64

1.3 Creating the postman profile

2.3.1Enter your email(name@learn-automation.com)

2.3.2 Username(candidate name with postman)

2.3.3 Password

2.3.4 Open next window

2.4 open new window

2.4.1 what is your name?

2.4.2 which is these roles is closets yours?

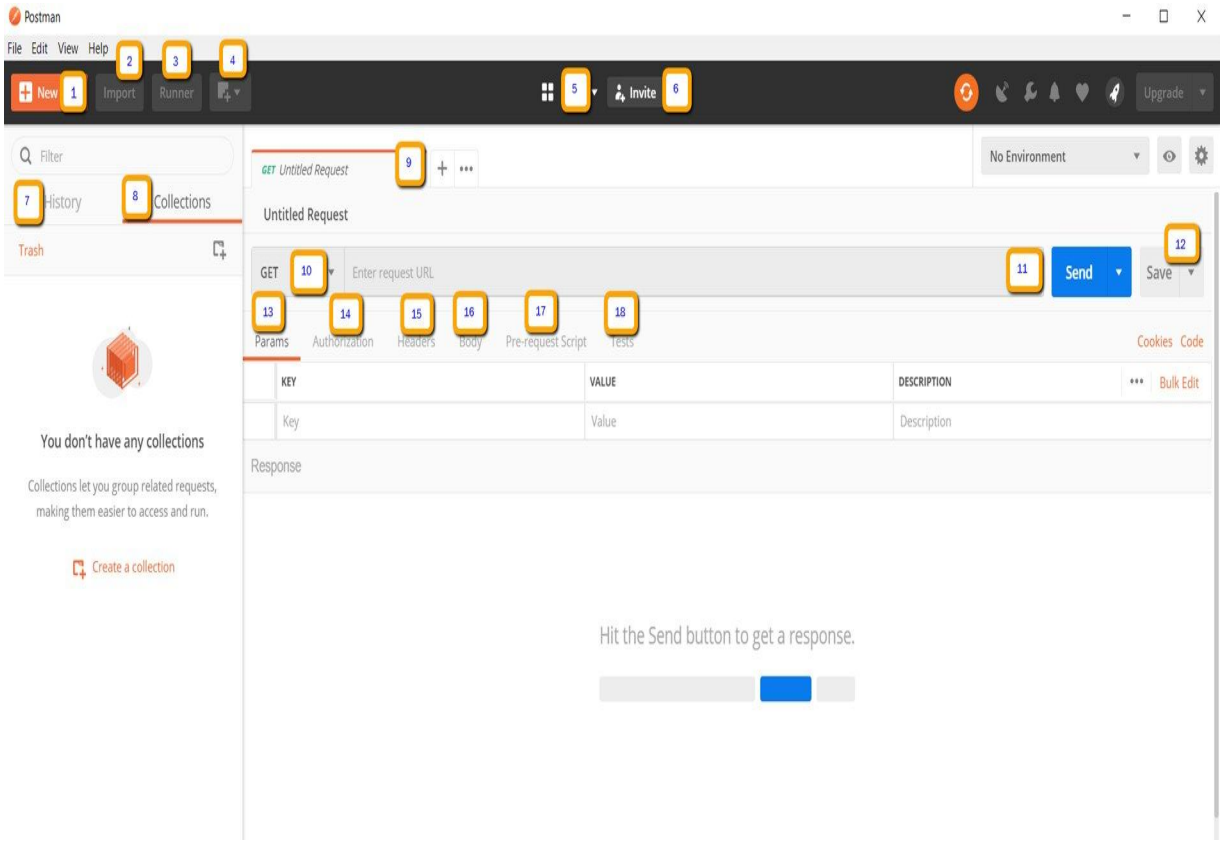
How do you plan to postman?

1.4 enter your team a name

1.5 Via a email

1.6 Continuous…. the open the postman

## 3.Postman navigation




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.

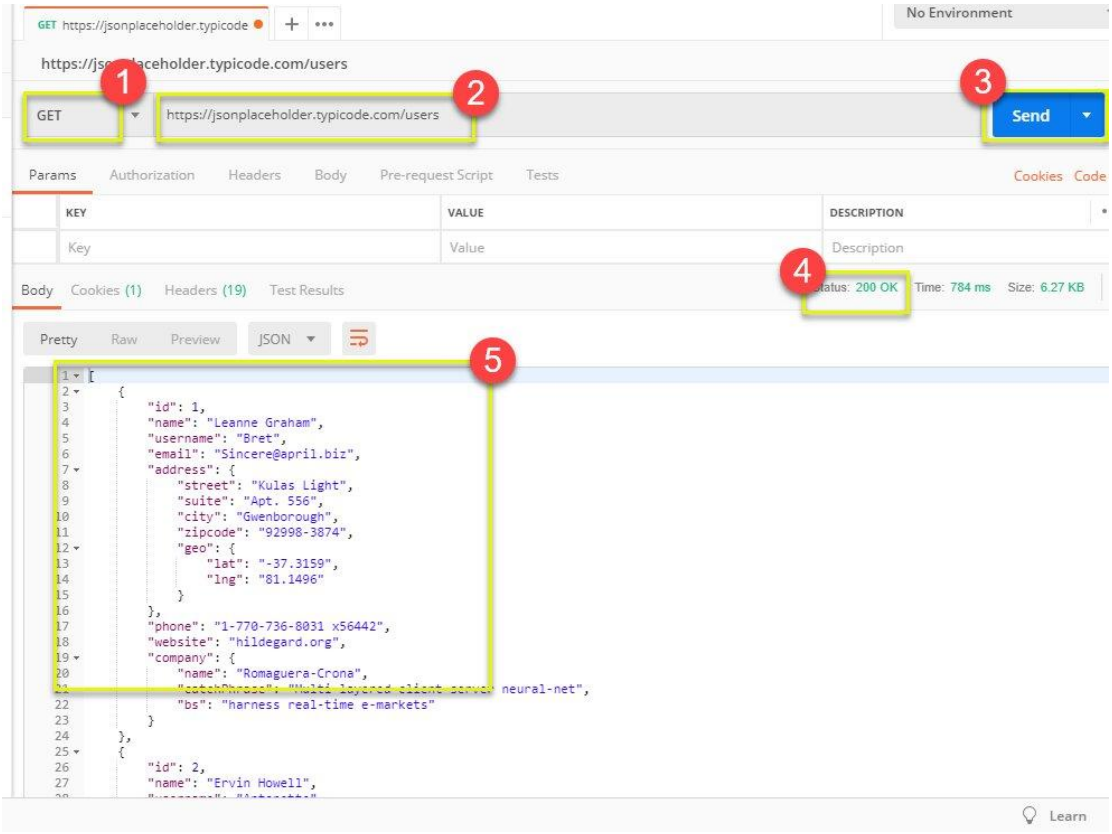
4. Open New – Open a new tab, Postman Window or Runner Window by clicking this button.
5. My Workspace – You can create a new workspace individually or as a team.
6. Invite – Collaborate on a workspace by inviting team members.
7. History – Past requests that you have sent will be displayed in History. 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 username and password, bearer token, etc.
15. Headers – You can set headers such as content type JSON depending on the needs of the organization.

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

**4.GET Request in postman**

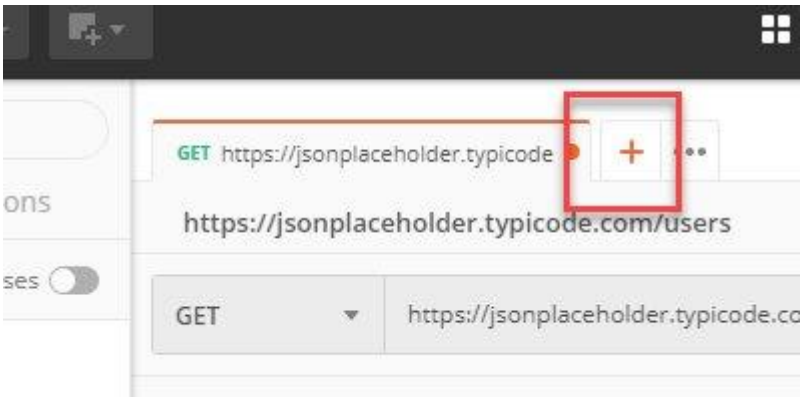
https://jsonplaceholder.typicode.com/users ( URL for all examples in this Postman tutorial)

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.




## 5.Post Request using postman

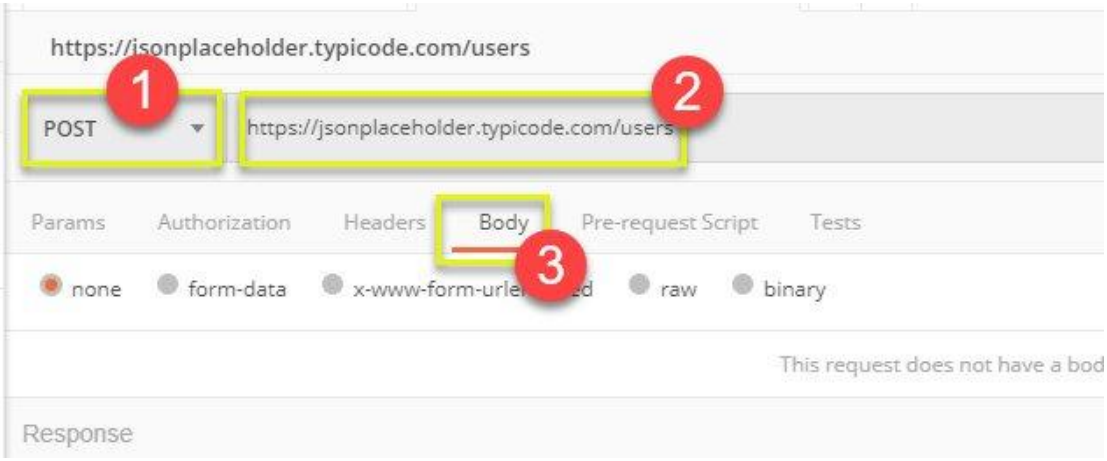
**Step 1)** Click a new tab to create a new request.




**Step 2)** In the new tab

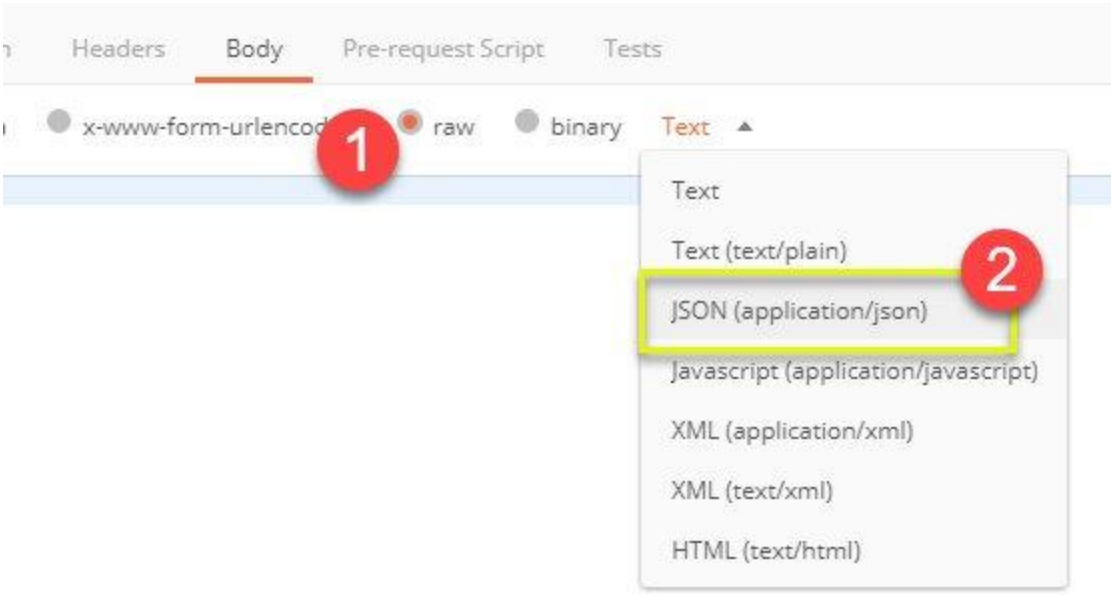
1. Set your HTTP request to POST.

2. Input the same link in request url: https://jsonplaceholder.typicode.com/users
3. switch to the Body tab




**Step 3)** In Body,

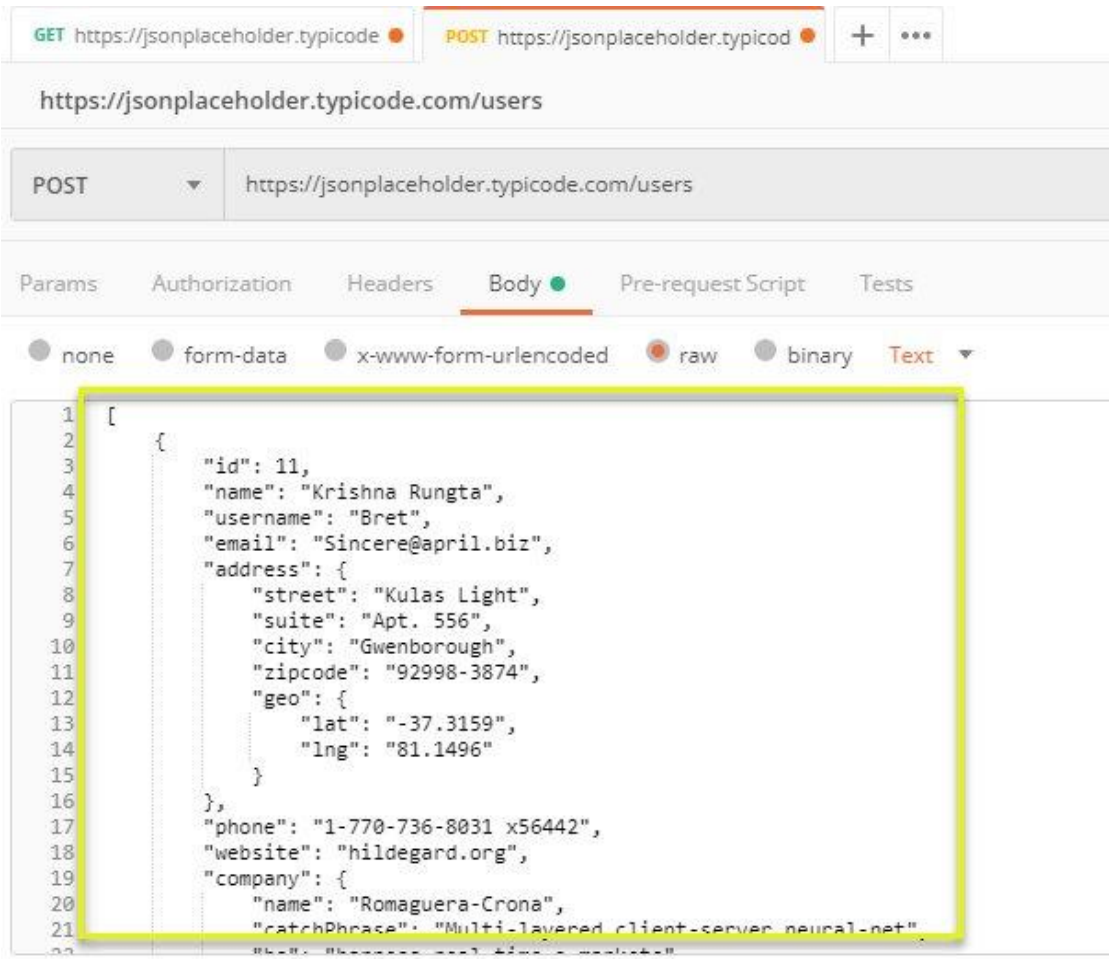
1. Click raw
2. Select JSON



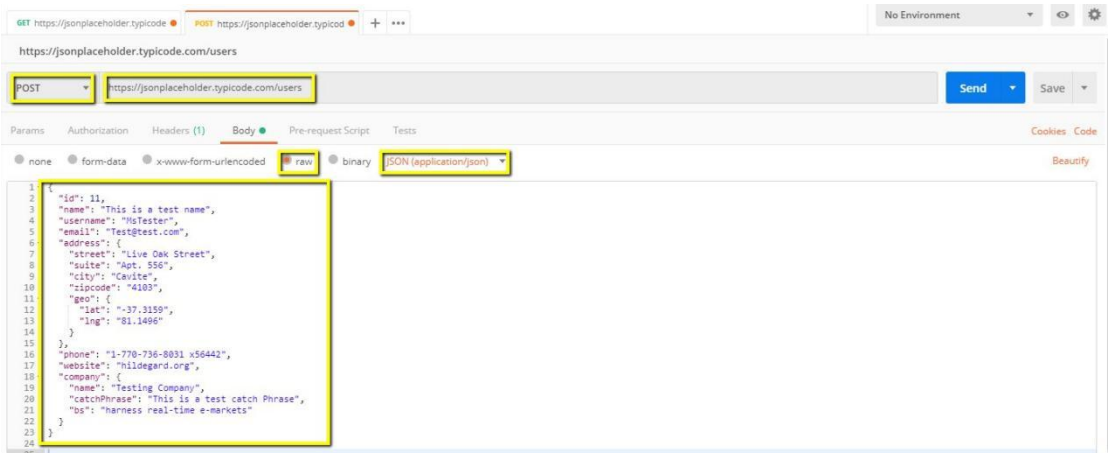
**Step 4)** Copy and paste just one user result from the previous get request like below. Ensure that the code has been copied correctly with paired curly braces and brackets. Change id to 11 and name to any desired name. You can also change other details like the address.


```
[
  {
    "id": 11,
    "name": "Krishna Rungta",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "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"
    }
  }
```

]

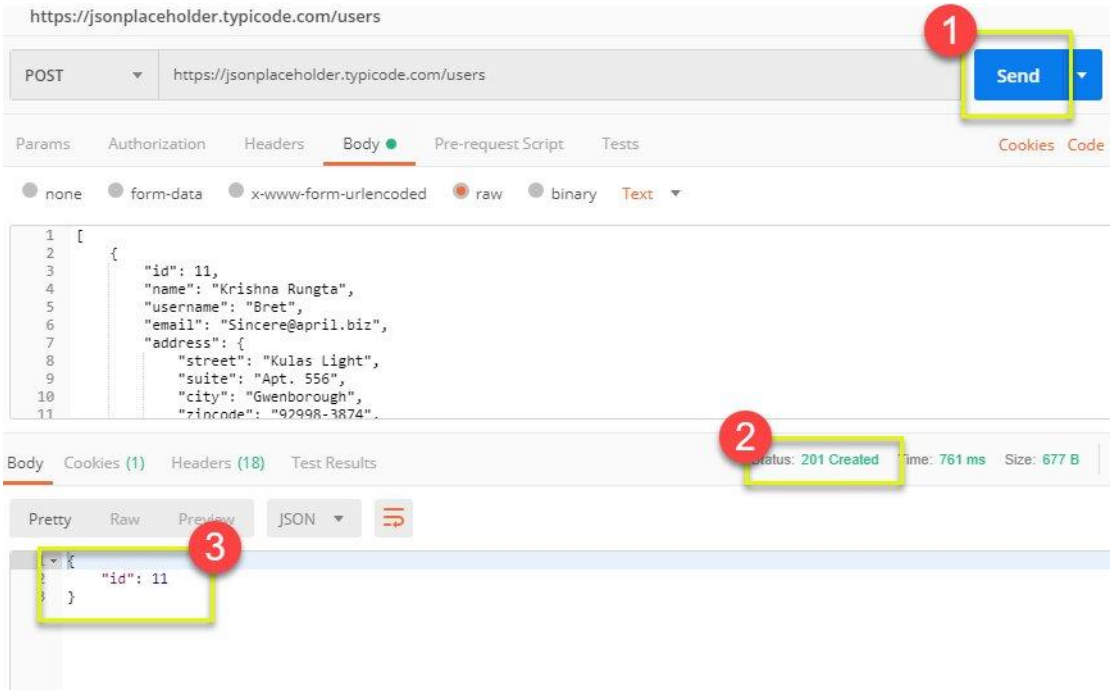



Online Post request should have the correct format to ensure that requested data will be created. It is a good practice to use Get first to check the JSON format of the request. You can use tools like https://jsonformatter.curiousconcept.com/




**Step 5)** Next,

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

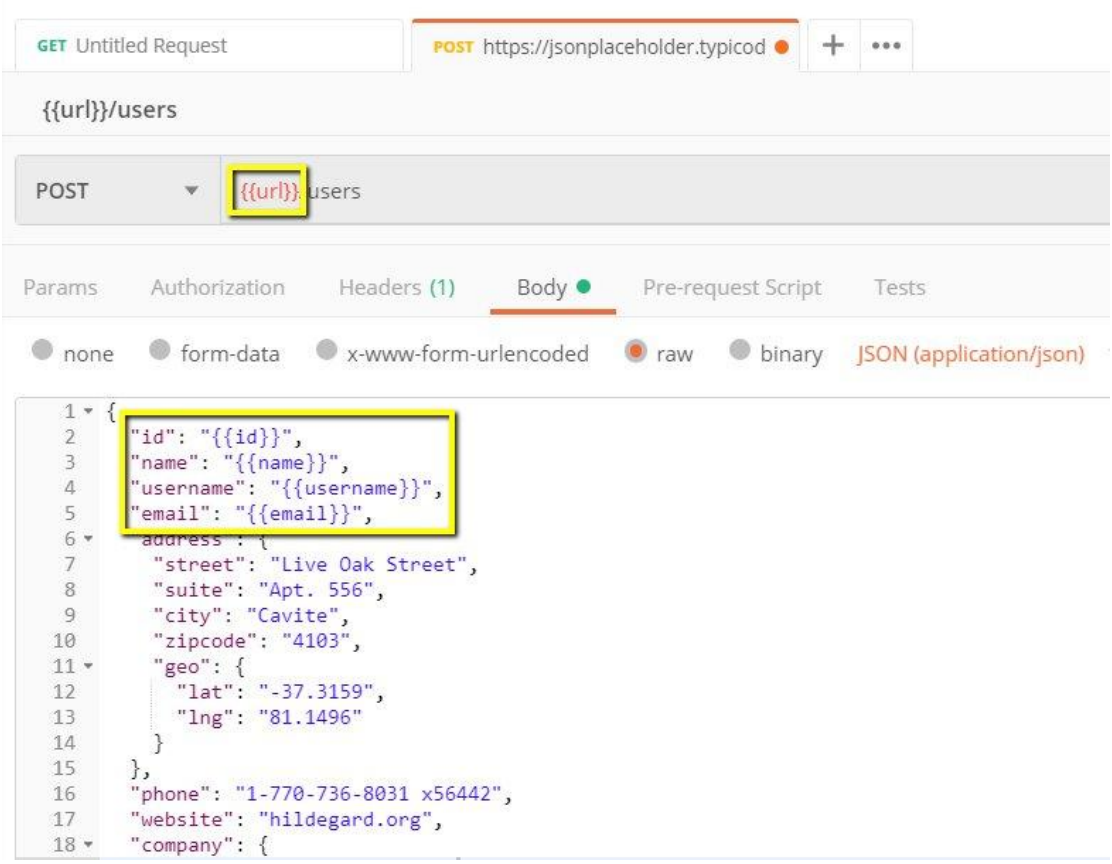



## 6.Request Parameters in postman

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.

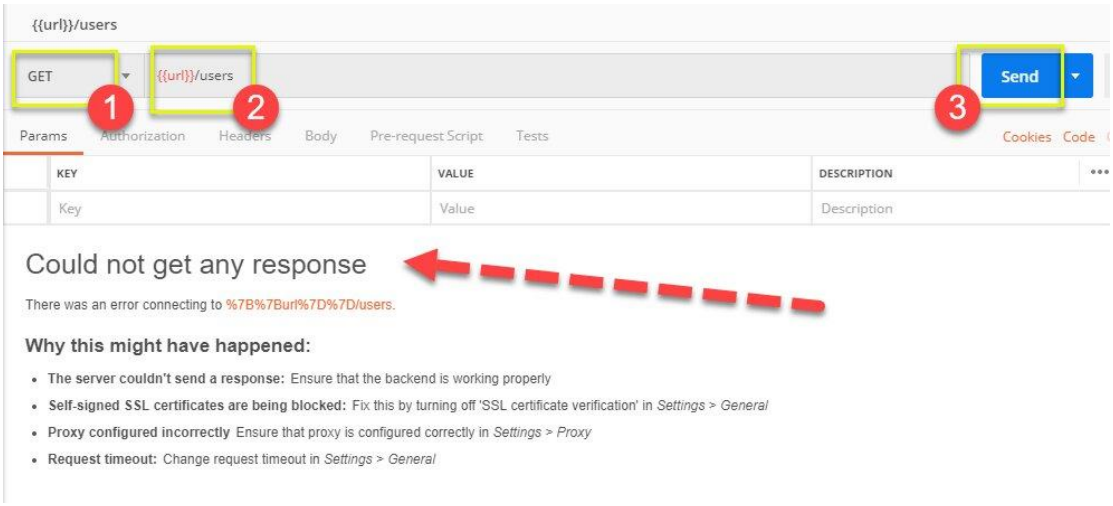
Parameters are created through the use of double curly brackets: {{sample}}. Let's take a look at an example of using parameters in our previous request:




**Step 1)**

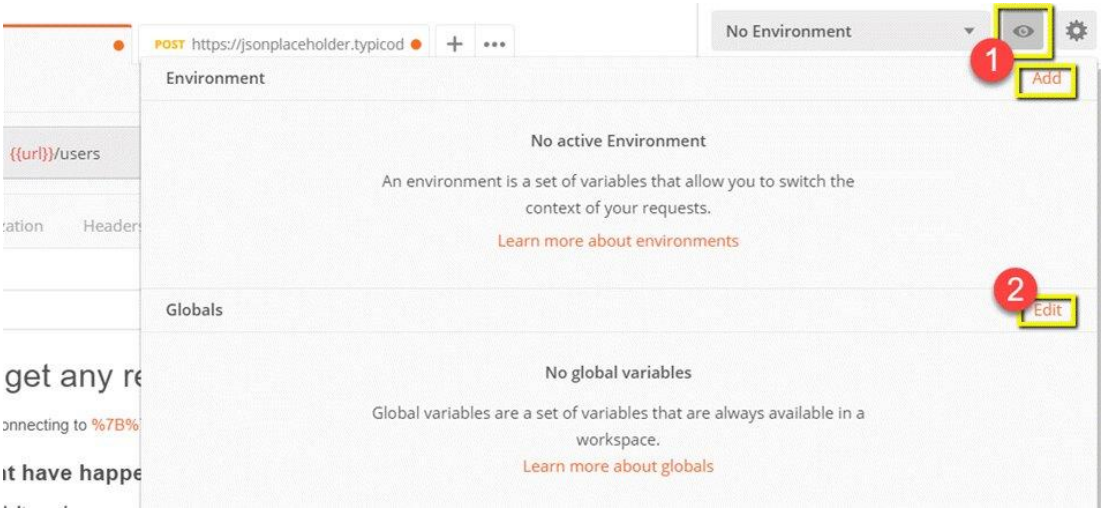
1. Set your HTTP request to GET
2. Input this link: https://jsonplaceholder.typicode.com/users. Replace the first part of the link with a parameter such as {{url}}. Request url should now be {{url}}/users.
3. Click send.

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.

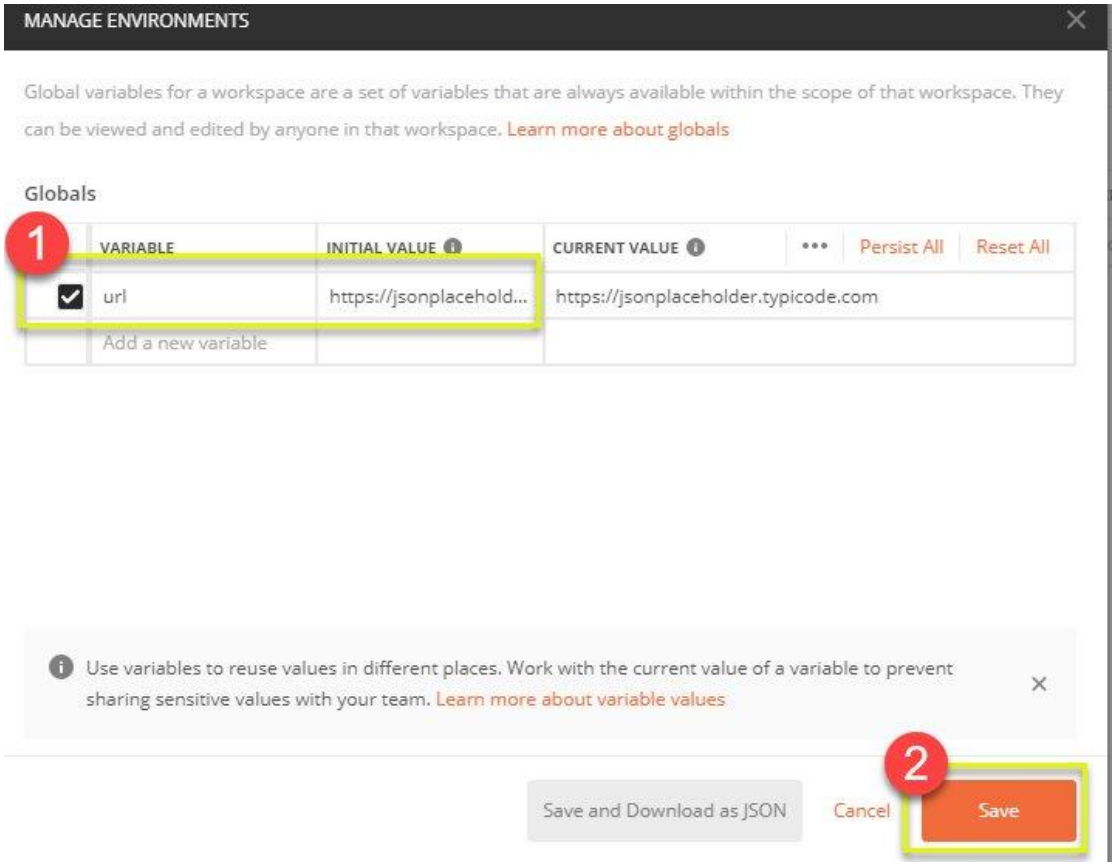



..**Step 3)** In variable,

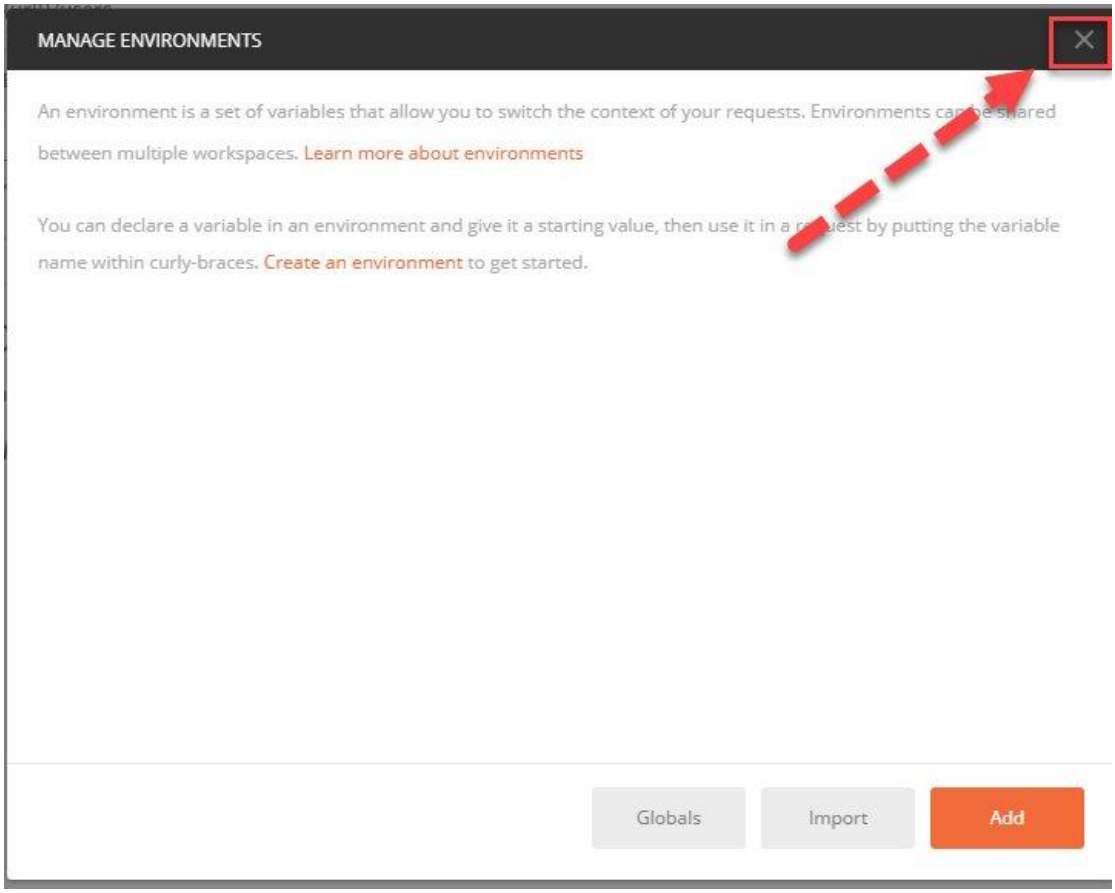
1. set the name to the url which is https://jsonplaceholder.typicode.com

2. click Save.

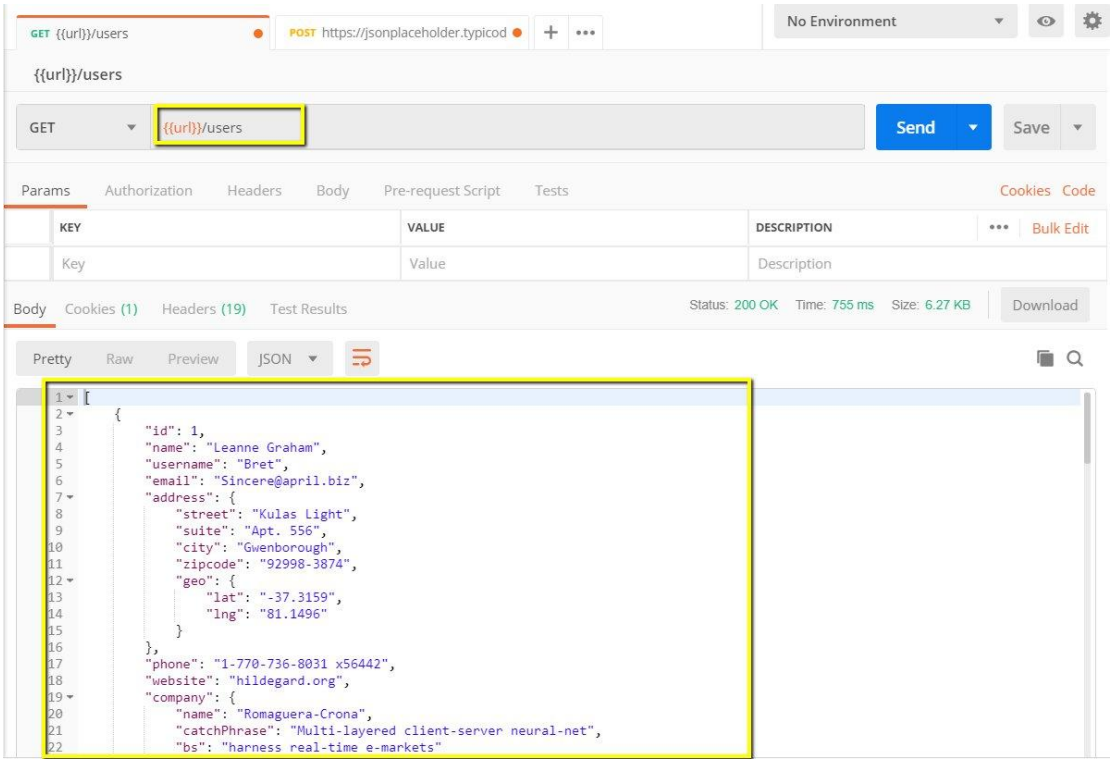
.




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



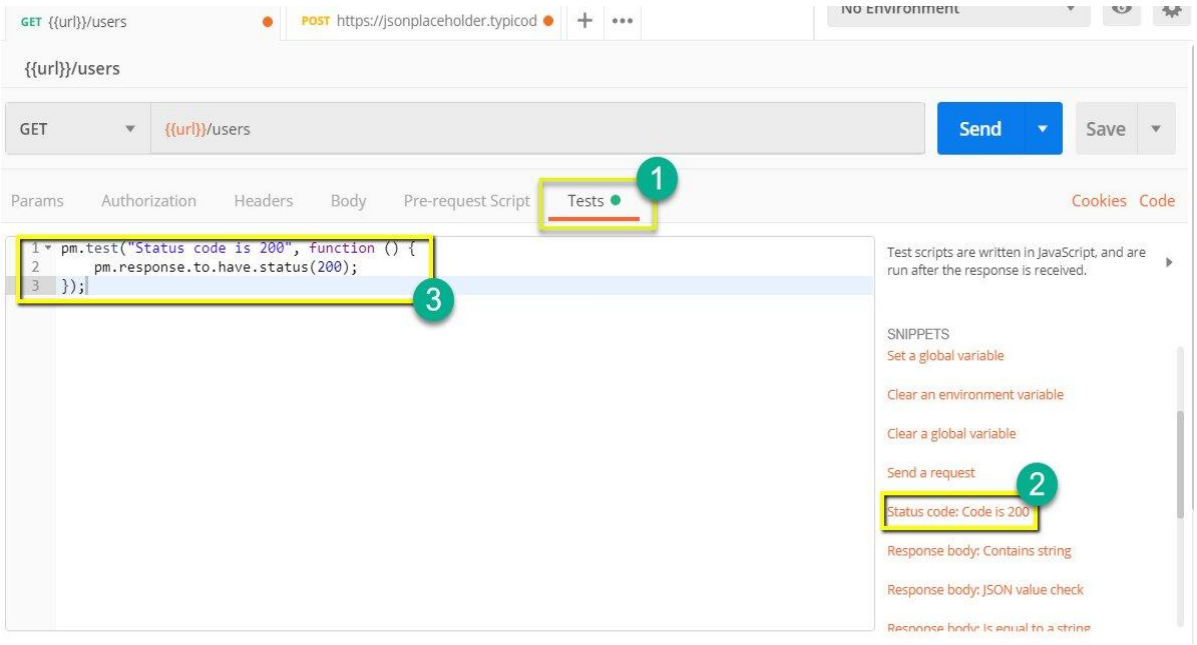

**Step 5)** Go back to your Get request then click send. There should now be results for your request.



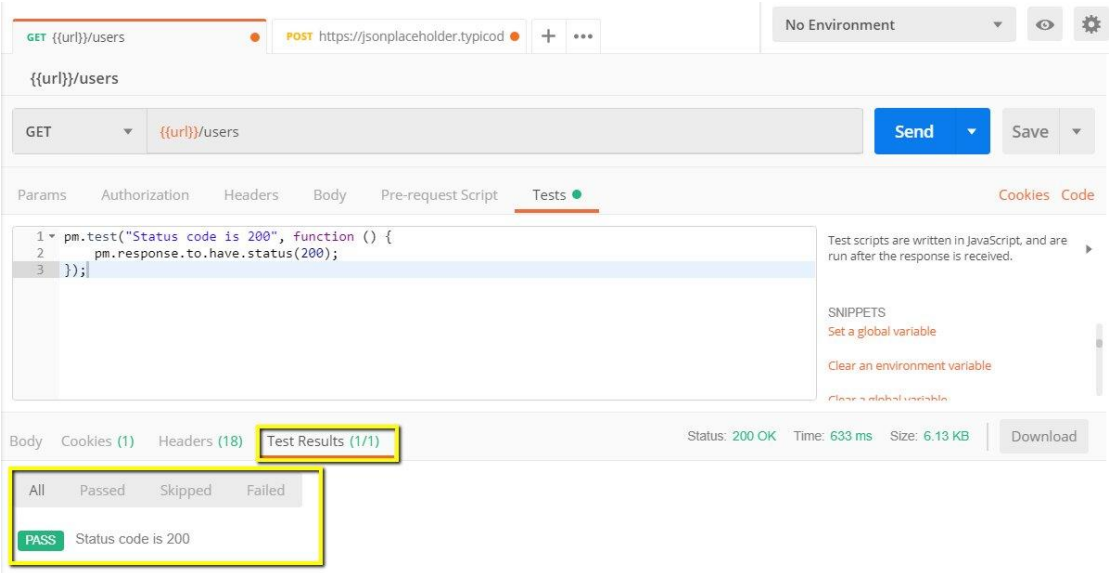

## 7.Create New Request in postman

**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”.




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



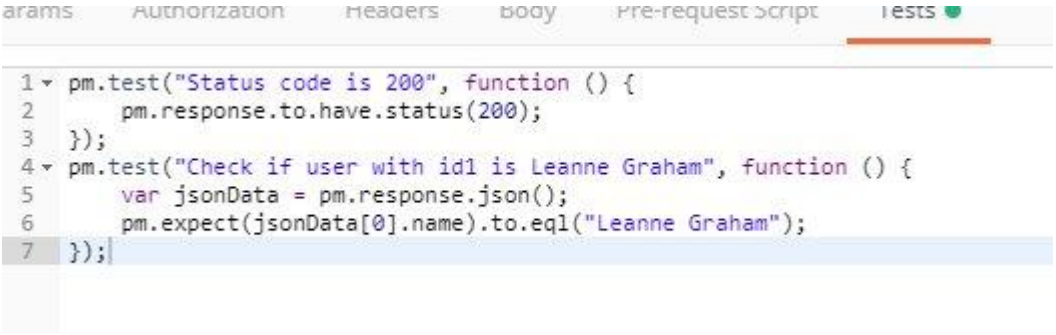

**Step 3)** Go back to the test tab and let's add another test. This time we will compare the expected result to the actual result.

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

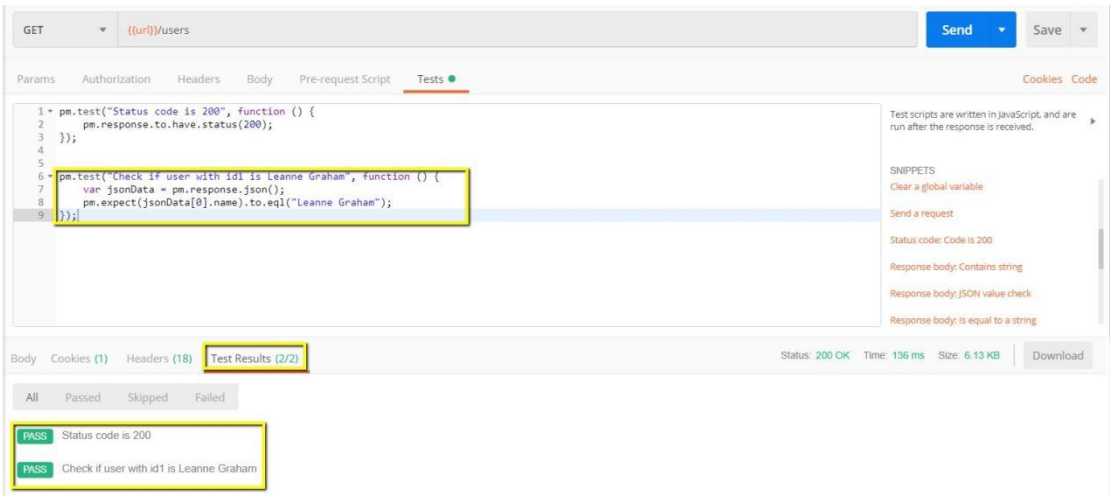

**Step 4)**

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");
});  ]
```

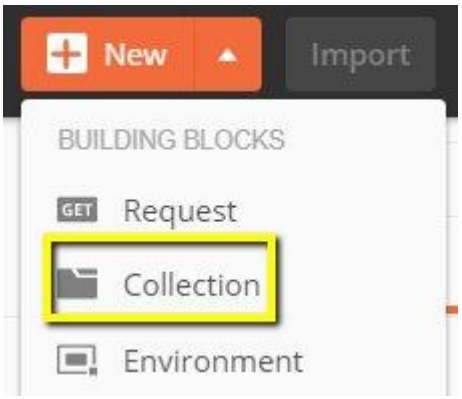



**Step 5)** Click send. There should now be two passed test results for your request.

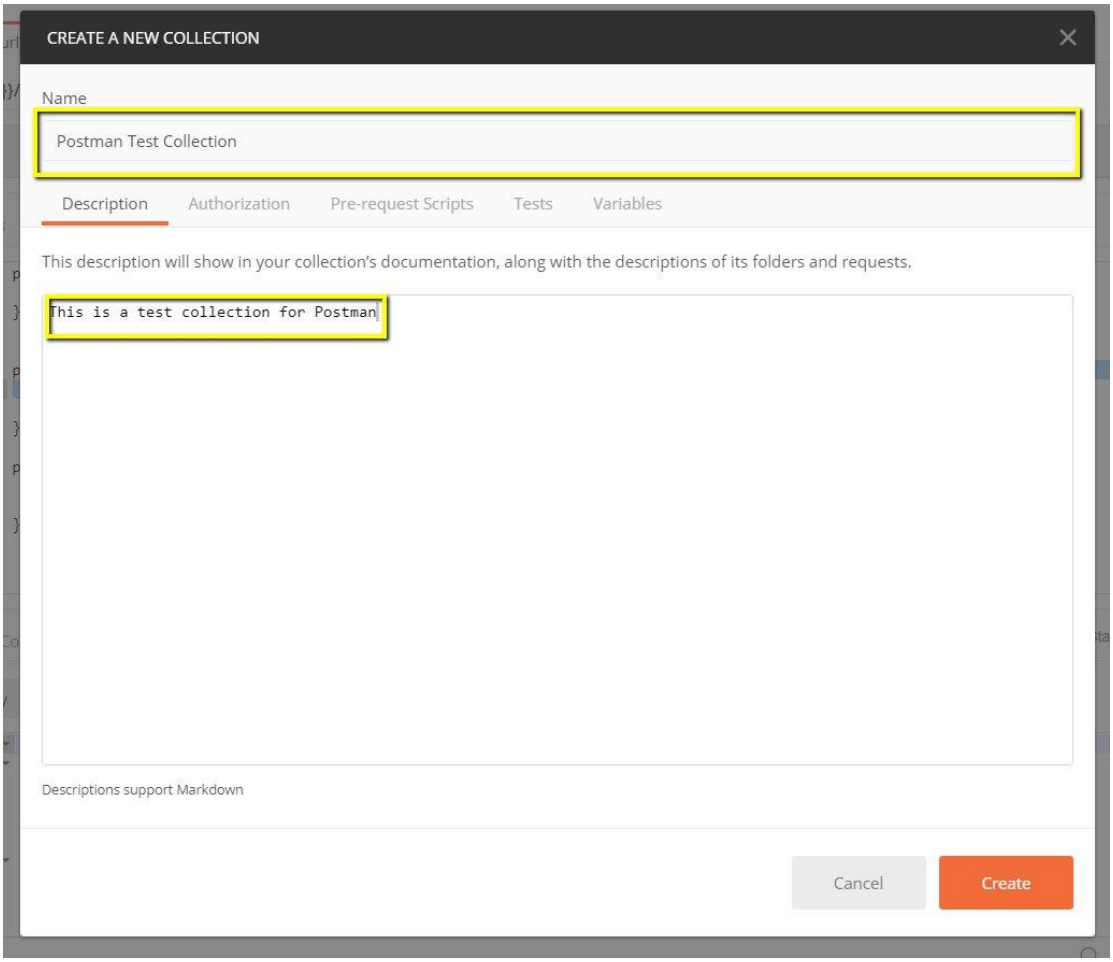



## 8.How to Create Collections

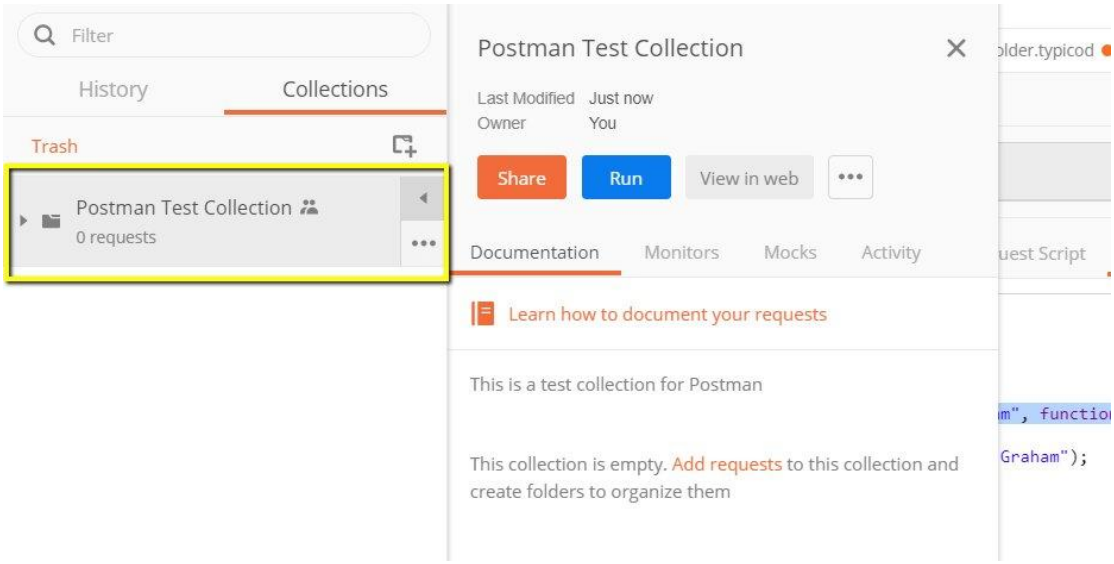
**Step 1)** Click on the New button at the top left corner of the page.



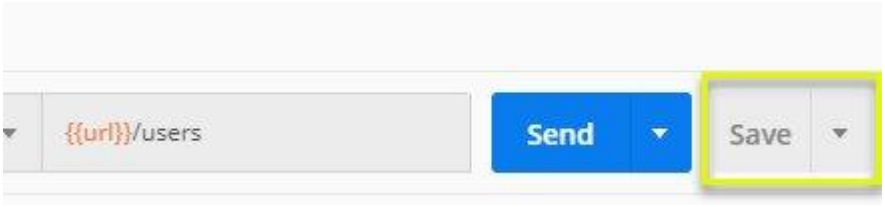
**Step 2)** Select Collection. Create collection window should pop up.




**Step 3)** Input the desired collection name and description then click create. A collection should now be created.

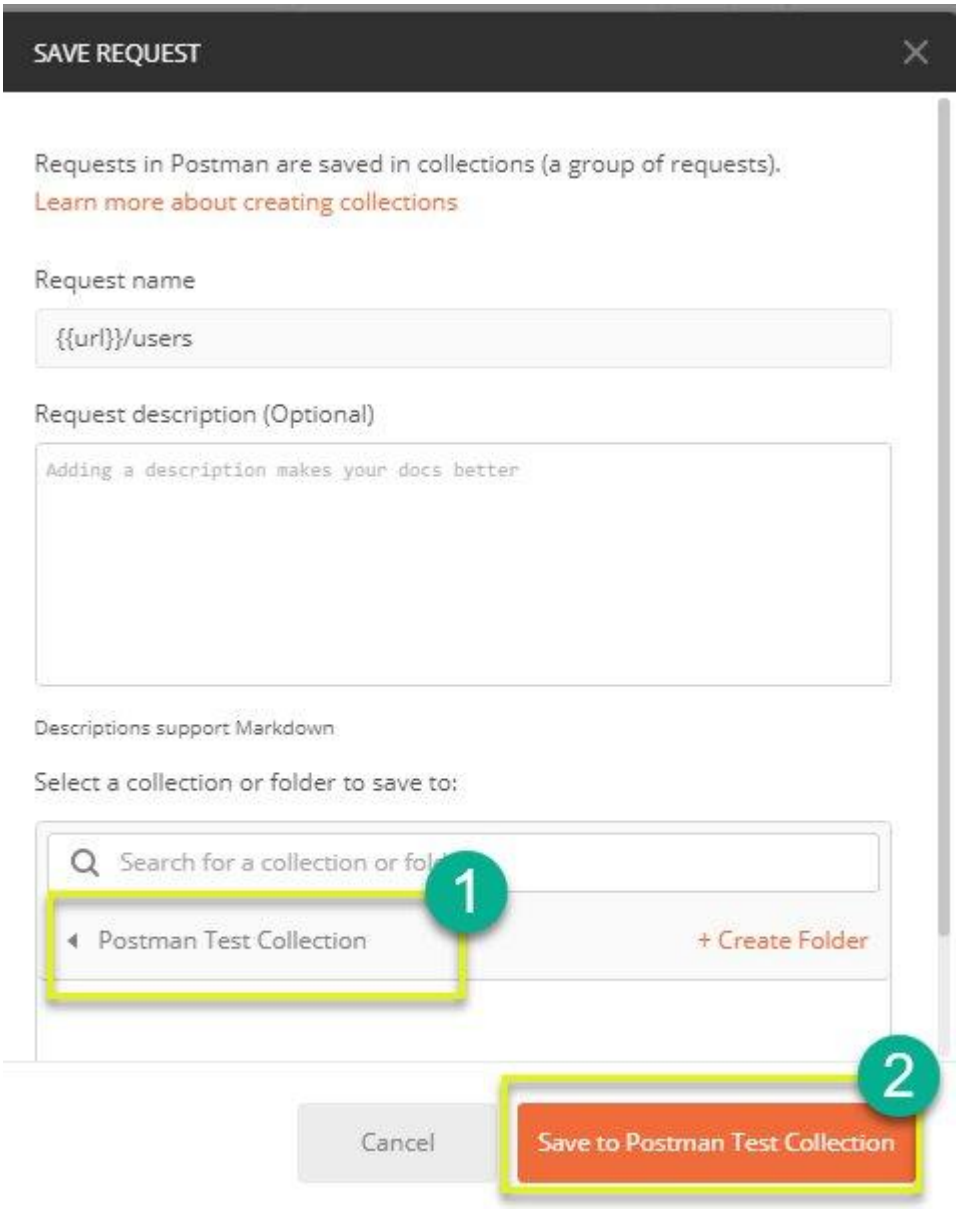



**Step 4)** Go back to the previous Get request. Click Save

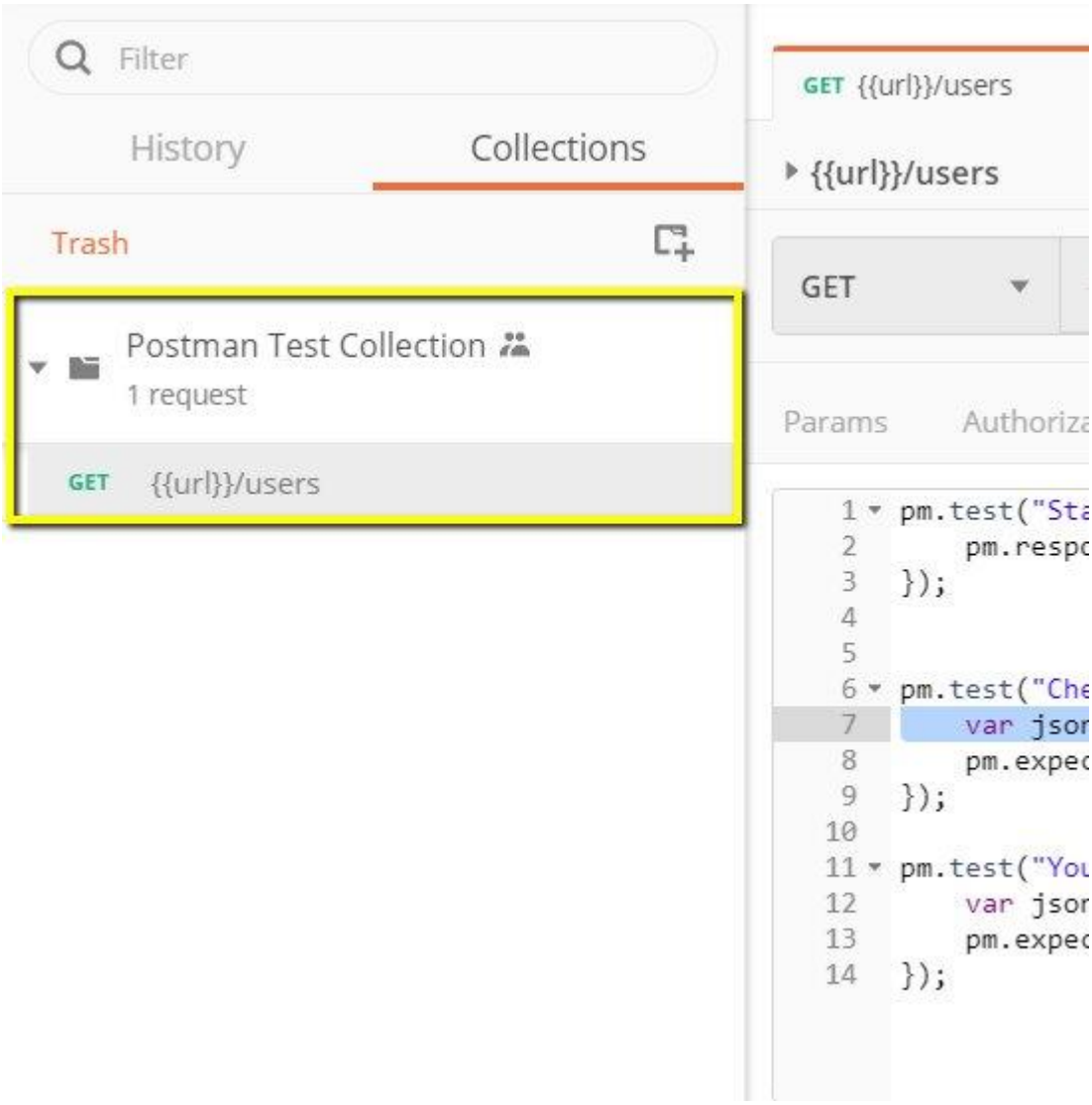



**Step 5)**

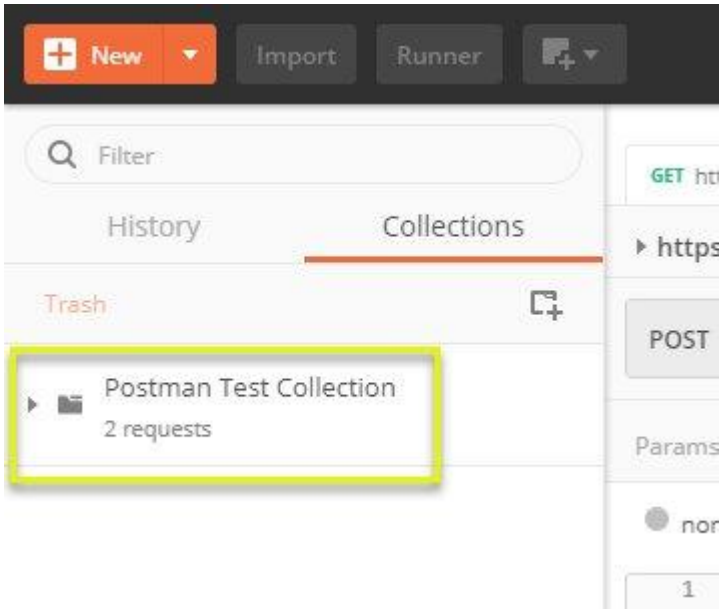
1. Select Postman Test Collection.
2. Click Save to Postman Test Collection




**Step 6)** Postman test collection should now contain one request.



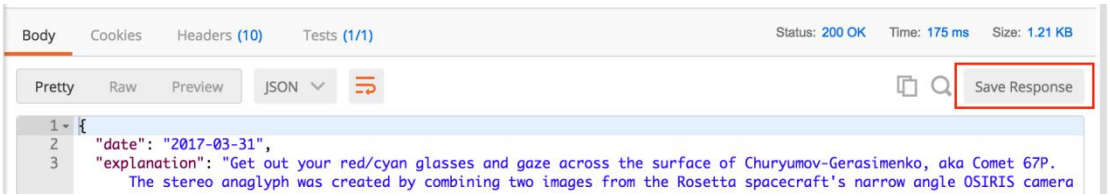

**Step 7)** Repeat steps 4-5 for the previous Post request so that collection will now have two requests.

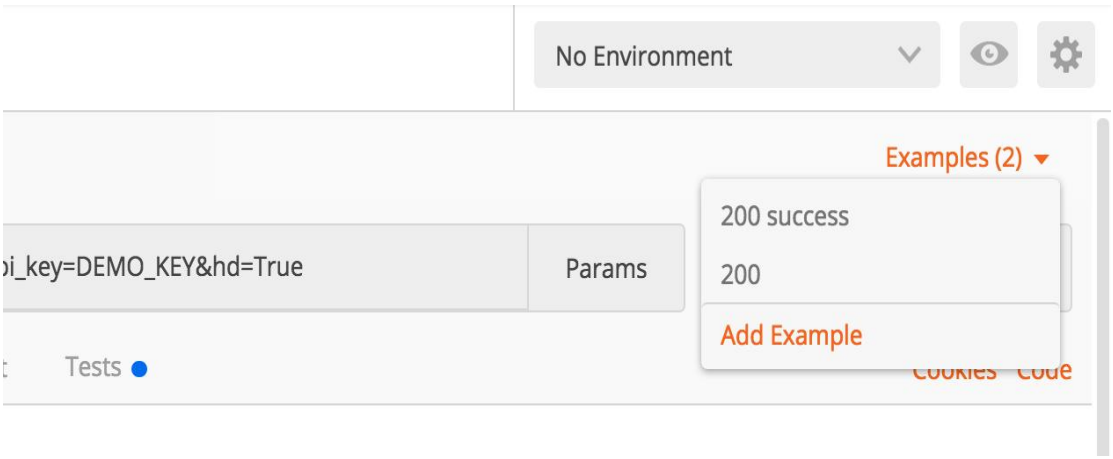



## 9. Response in postman

An API response consists of the body, headers, and the status code. Postman organizes body and headers in different tabs. The status code with the time taken to complete the API call is displayed next to the tabs. You can hover over the status code to get more details about the code. Mostly it will be the default description as mandated by the HTTP specification, however, API authors can also add custom messages.

**Saving responses**

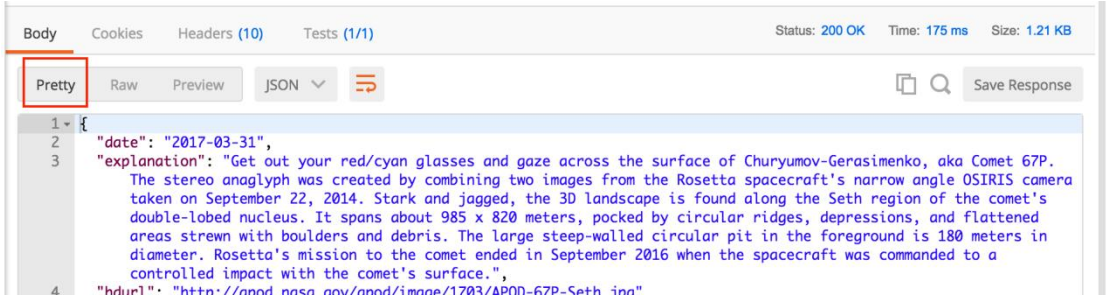



## Viewing responses

The Postman **Body** tab gives you several tools to help you make sense of things quickly. The body can be viewed in one of three views - pretty, raw, and preview.

## Pretty

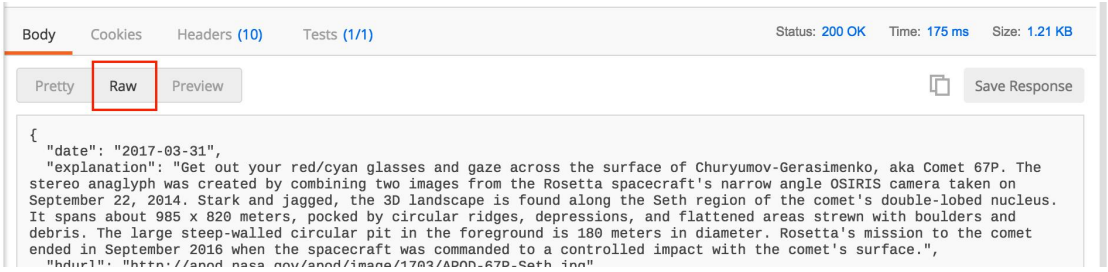


The pretty mode formats JSON or XML responses so that they are easier to look at. Nobody wants to scroll through a minified single line JSON response looking for that elusive string! Links inside the pretty mode are highlighted and clicking on them can load a GET request in Postman with the link URL. For navigating large responses, click on the down-pointing triangles (▼) on the left to collapse large sections of the response.

For Postman to automatically format the body, make sure the appropriate Content-Type header is returned. If the API does not do this, then you can force formatting through JSON or XML. You can force JSON formatting under the **General** tab within the **SETTINGS** modal by selecting "JSON" from the "Language detection" dropdown.

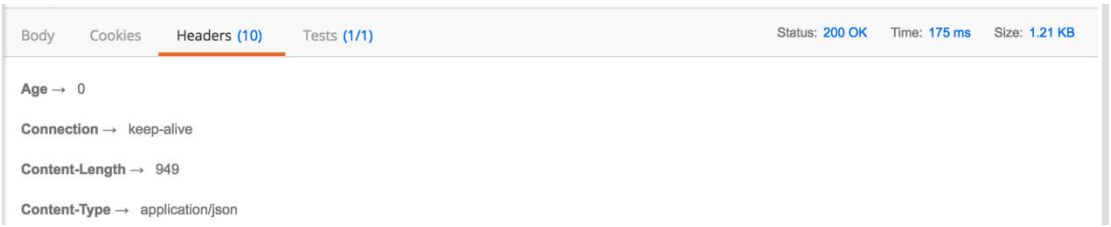
**Finding items in responses:** You can use **CMD/CTRL + F** to open the search bar, and **CMD/CTRL + G** to scroll through results. See complete set of keyboard shortcuts.

**Raw**




he raw view is just a big text area with the response body. It can help to tell whether your response is minified or not.

**Preview**




**Response time**

Postman automatically calculates the time it took for the response to arrive from the server. This is useful for some preliminary testing for performance.

**Response size**

Postman breaks down the response size into body and headers. The response sizes are approximate.

**Cookies**

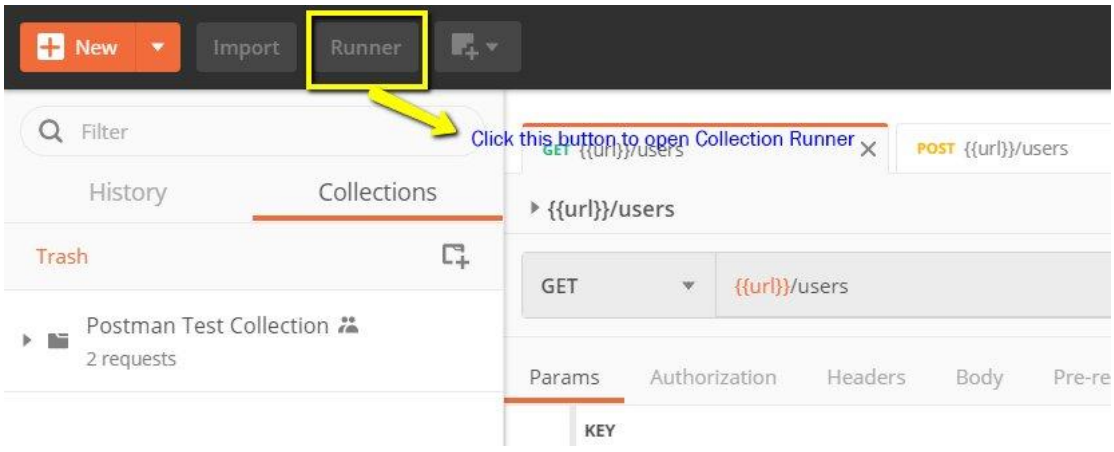
Cookies sent by the server are visible in a dedicated tab. To manage cookies in Postman the native apps, use the **MANAGE COOKIES** modal. If you're working in the Postman Chrome app, you can use the Interceptor extension to help manage cookies.

**Tests**

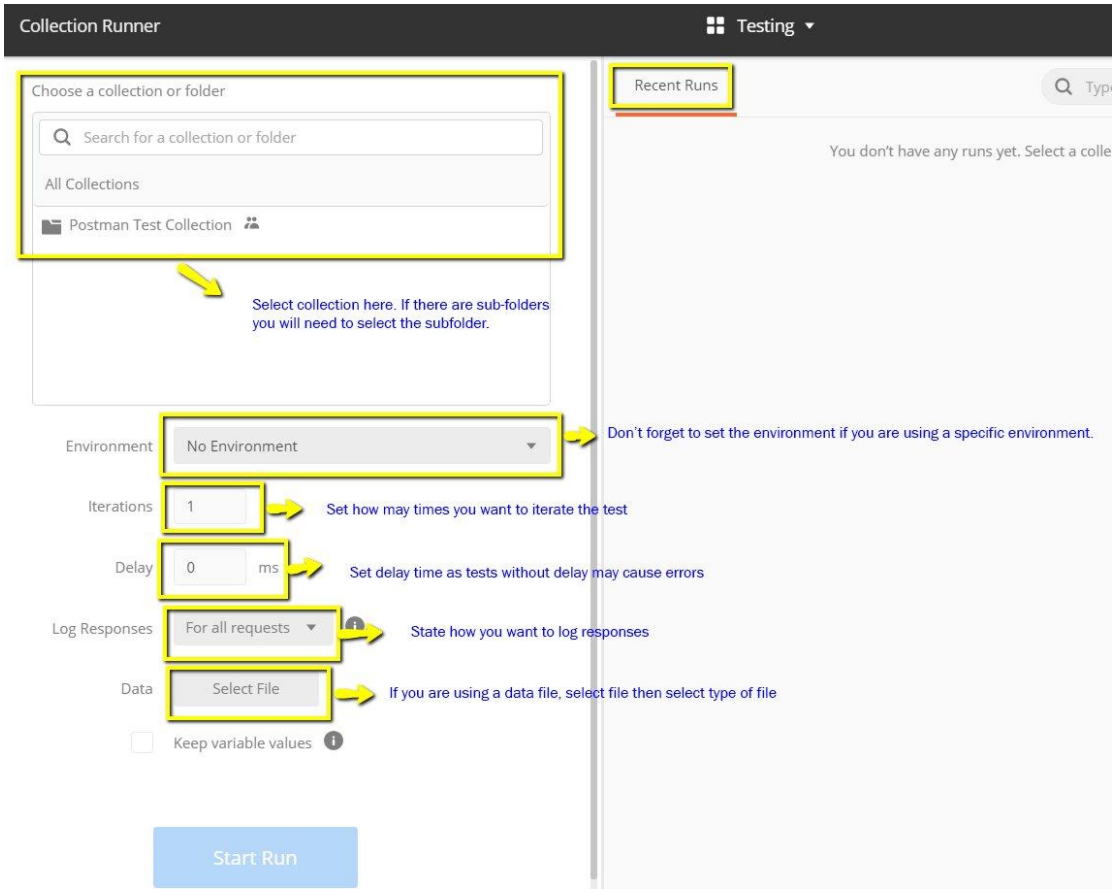
Along with everything that you get from the server for the request, you can also see the results of the tests that were run against the request. Learn more about testing in Postman.

## 10.How to Run Collections using Collection Runner

**Step 1)** Click on the Runner button found at the top of the page next to the Import button.

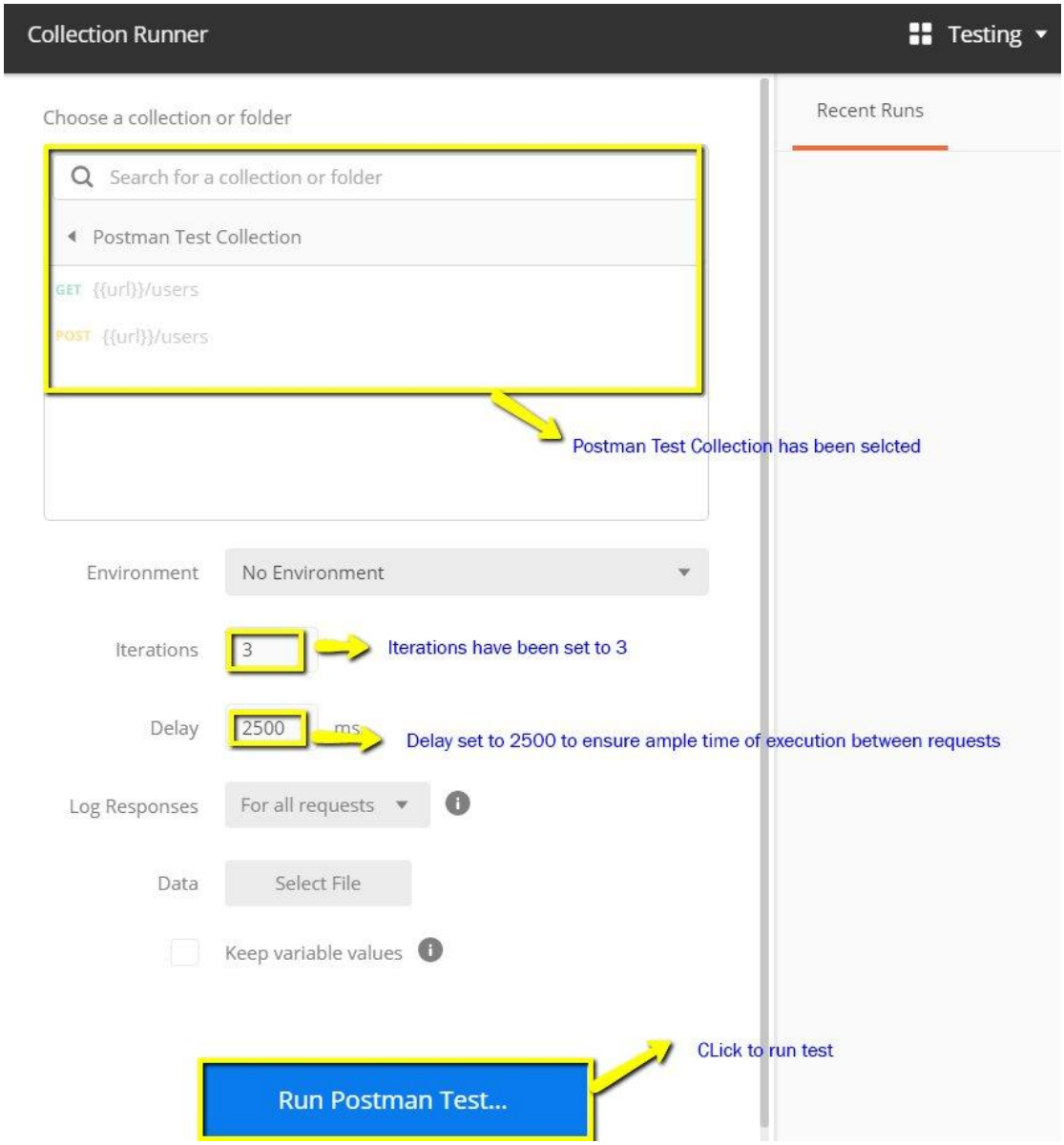


**Step 2)** Collection Runner page should appear such as below. Following is the description of various fields



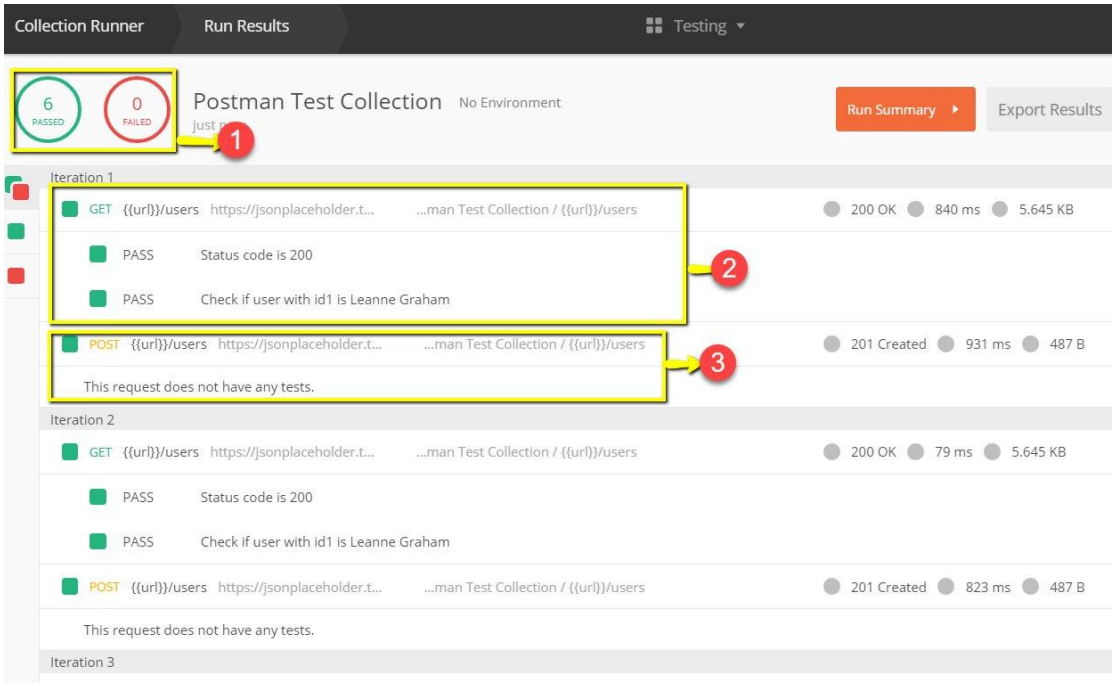

**Step 3)** Run your Postman Test Collection by setting up the following:

- Choose Postman test collection- Set iterations as 3.
- Set delay as 2500 ms
- Click on Run Postman Test… button




**Step 4)** Run Results page should be displayed after clicking the Run button. Depending on the delay, you should see the tests as they execute.

1. Once tests have finished, you can see the test status if it is Passed or Failed and the results per iteration.
2. You see Pass status for the Get Requests
3. Since we did not have any tests for Post, there should be a message that the request did not have any tests.




You can see how important it is that there are tests in your requests so that you can verify HTTP request status if successful and the data is created or retrieved.

- API Testing using Postman: Postman is an application for testing APIs. Postman is one of the most popular tools used in API testing by sending requests to the webserver and getting the response back
- Accessibility, Use of Collections, Collaboration, Continuous Integration, are some of the Key features to learn in Postman
- It’s recommended you create an account in Postman, so your collections are available online
- You can parameterize request in Postman
- You can create Tests to verify a postman request
- Collections can be run using Newman or Collection Runner