

Introduction to API Testing

Lesson 3: Postman Advance Concepts

Lesson Objectives

- Basic Authentication in Postman
 - Need of Authentication
 - Authentication Vs Authorization
 - Basic Authentication in Postman
- Environmental Variables in Postman
 - What is an Environment in Postman?
 - What is an Environment Variable in Postman?
 - How to Create an Environment in Postman
 - How to Use Environment Variables in Postman
 - Scope of Variables in Postman
- Collections in Postman
 - Creating Collections in Postman
 - Run Collections using Collection Runner
- Assertions in Postman
 - What is assertion



Lesson Objectives

- Assertions in Postman
 - Why we use Assertion in Postman
 - Example
- Summary



3.1: Basic Authentication in Postman

Need of Authorization

- Whereas ***Authorization*** is a process of allowing or denying someone from accessing something, once ***Authentication*** is done.
- Authorization plays a very important role in deciding the accesses and tightening the security

3.1: Basic Authentication in Postman

Authentication Vs Authorization

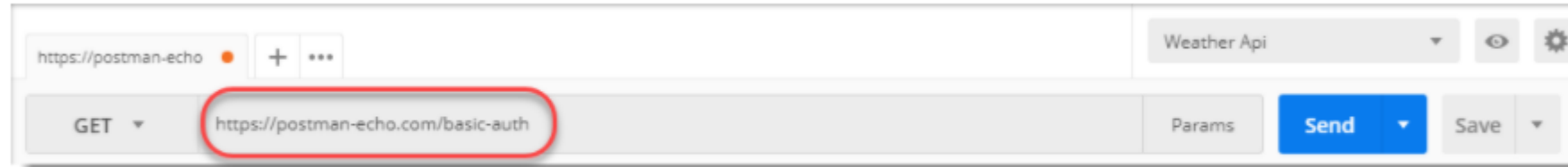
- **Authorization** and **Authentication** are two closely related terms. These two terms can also be confusing at first.

| Authentication | Authorization |
|--|---|
| Determines whether users are who they claim to be | Determines what users can and cannot access |
| Challenges the user to validate credentials (for example, through passwords, answers to security questions, or facial recognition) | Verifies whether access is allowed through policies and rules |
| Usually done before authorization | Usually done after successful authentication |
| Generally, transmits info through an ID Token | Generally, transmits info through an Access Token |
| Generally governed by the OpenID Connect (OIDC) protocol | Generally governed by the OAuth 2.0 framework |
| Example: Employees in a company are required to authenticate through the network before accessing their company email | Example: After an employee successfully authenticates, the system determines what information the employees are allowed to access |

3.1: Basic Authentication in Postman

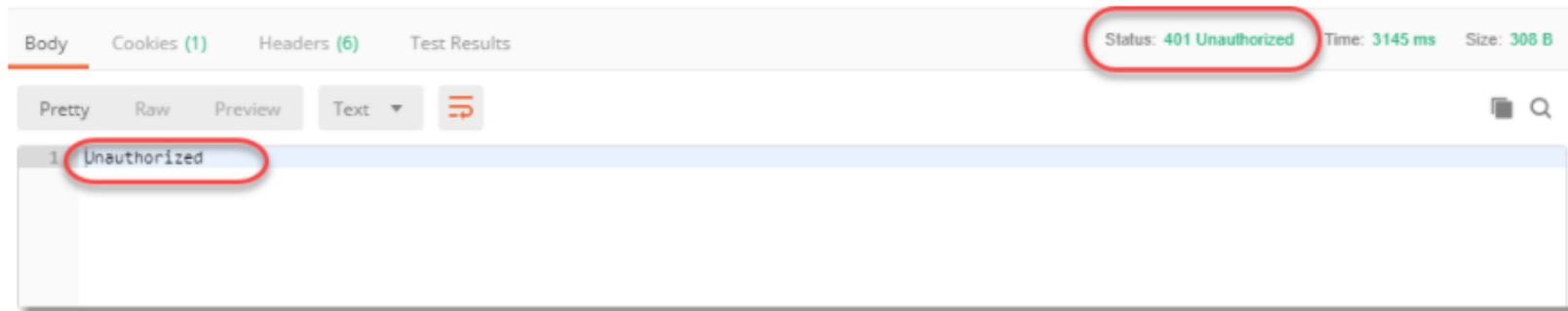
Authorization using Postman

- Create a *GET* request and enter the endpoint as ***https://postman-echo.com/basic-auth***



- Press ***send*** and look at the response

Press ***send*** and look at the response

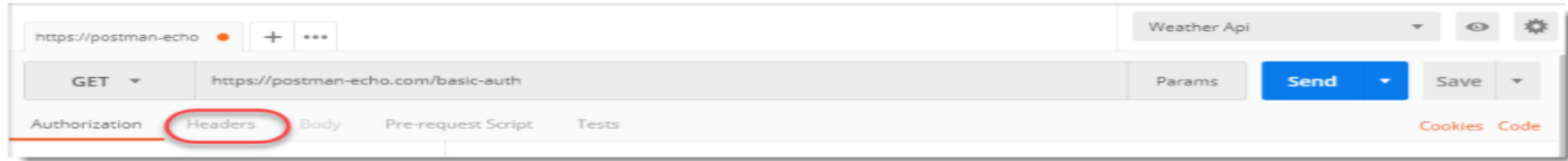


- If the status code is ***401*** which corresponds to unauthorized access and the response message says ***Unauthorized***.

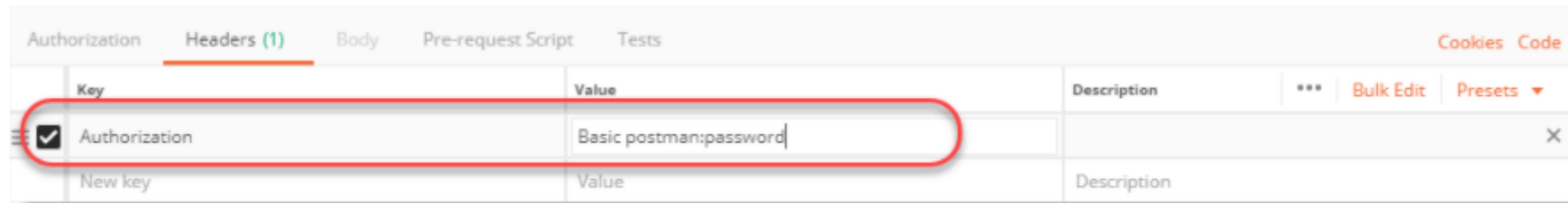
3.1: Basic Authentication in Postman

Basic Access Authentication

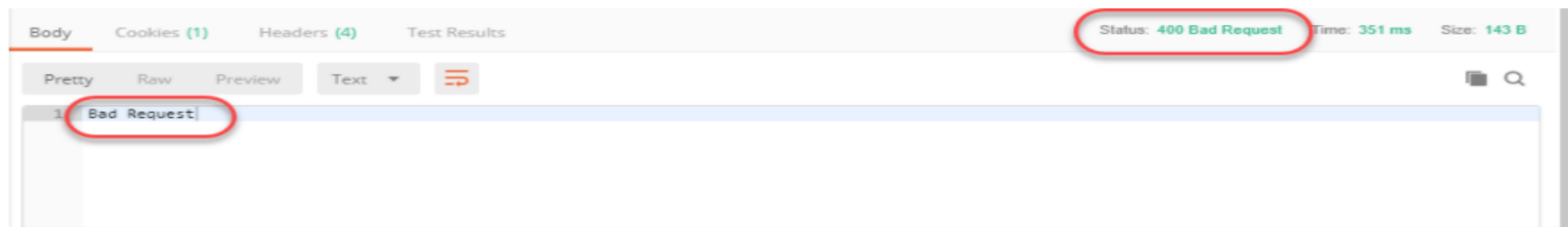
- Enter the endpoint ***https://postman-echo.com/basic-auth*** in **GET** request.
- Go to **Headers**



- Enter the following key value pairs in Header



- Use username as ***postman*** and password as ***password*** and Press ***Send*** and see the response box and status code.



3.2: Environmental Variables in Postman

What is an Environment in Postman?

- An environment in Postman is a ***set of key-value pairs***.
- It helps us to differentiate between the requests.
- With the creation of environment inside Postman, the change of key value pair occurs and these changes are reflected in the request.
- Environment provides boundaries to variables.
- On creating the environment, we can make track of all the variables and how to use them inside our requests can be achieved.

3.2: Environmental Variables in Postman

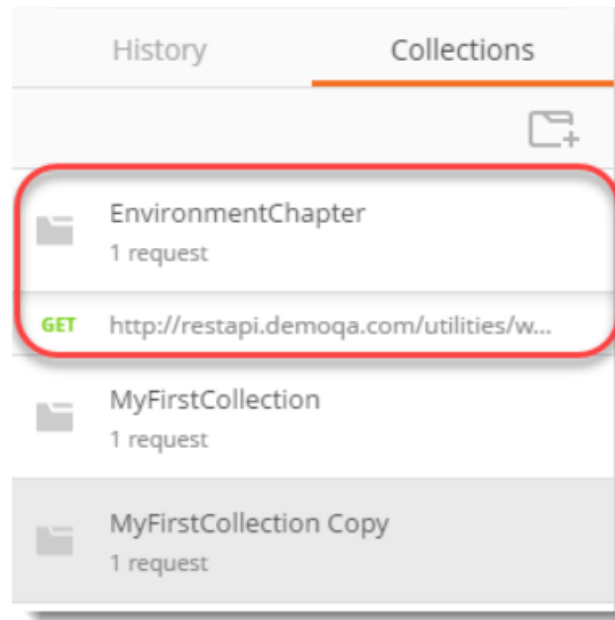
What is an Environment Variable in Postman?

- A variable is an entity whose value can be changed.
- ***The key part in the key-value set in environment is called variable.***
- This variable can have any value and in place of the ***key*** we can use the variable name in every request.
- Environment Variable in Postman can be created by using following three steps in the process:
 - *Create an Environment*
 - *Create Environment Variables*
 - *Use an Environment Variable in the Request*

3.2: Environmental Variables in Postman

How to Create an Environment in Postman

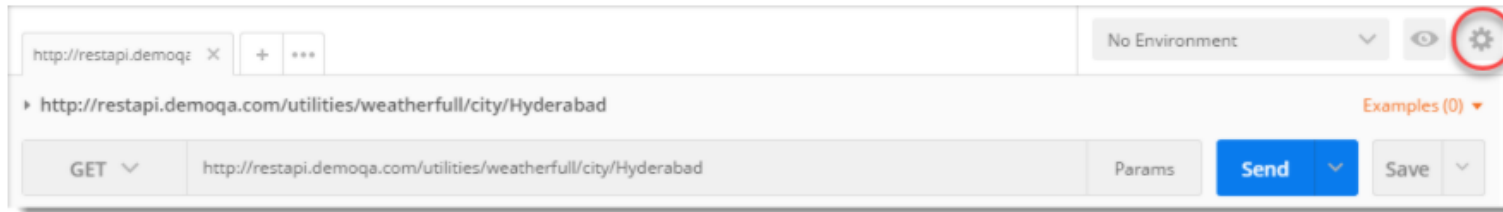
- Create a new Collection and name it as ***Environment_Chapter***.
- Add Weather Api Request in the collection used in the Get Request chapter.



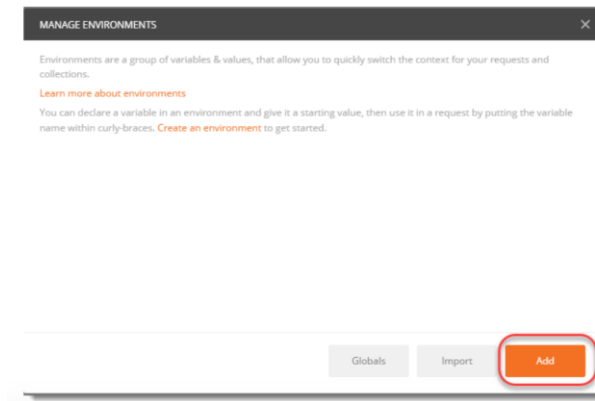
4.2: Environmental Variables in Postman

How to Create an Environment in Postman Continue...

- Click on the **gear** icon which says Manage Environment.



- Click on **Add**.



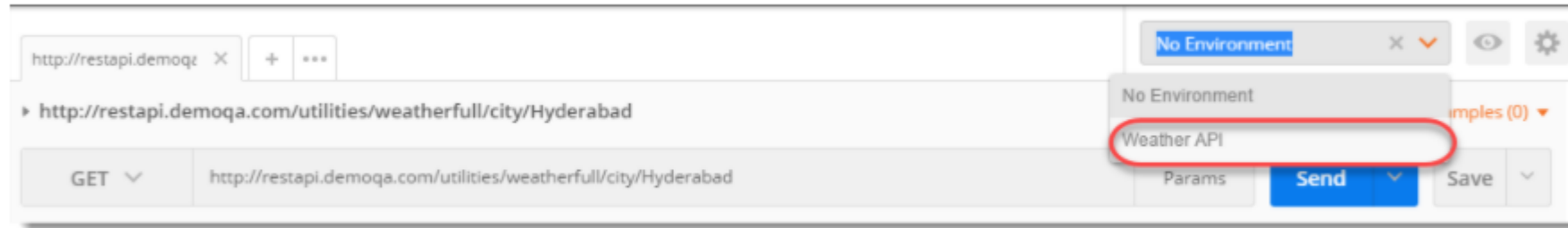
- Name the environment as **Weather API**
- Click on **Add** and close the panel.



3.2: Environmental Variables in Postman

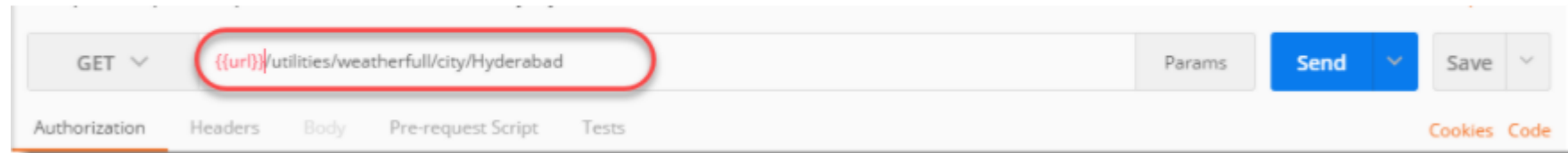
How to Use Environment Variables in Postman

- Select the dropdown which says No Environment and select **Weather API** environment in that.



- In the address bar change **http://restapi.demoqa.com** to **{{url}}**

2. In the address bar change **http://restapi.demoqa.com** to **{{url}}**



- Click on **Send**. Now, we have created an environment and used a variable called **URL** here.

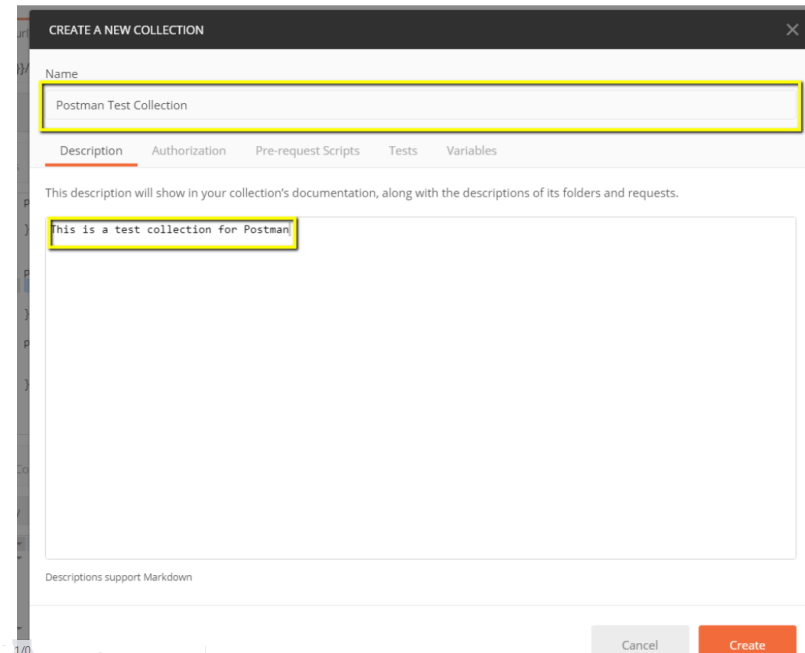
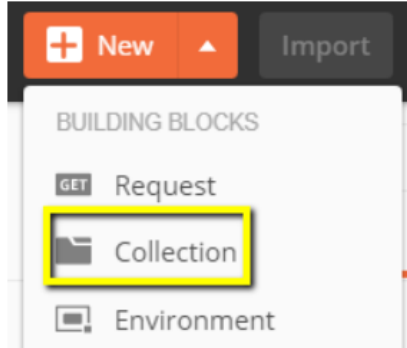
3.2: Environmental Variables in Postman

Scope of Variables in Postman

- A scope of anything is the ***boundary in which that thing can be accessed and perform.***
- variables in Postman have two scopes
 - ***Local Scope*** : *Local Scope Variables* can only work inside the environment in which it was created. *Changing the environment will stop the access to that variable and we will encounter an error.*
 - ***Global Scope*** : The variable URL that we just created above is the local variable because it has a scope only till the environment Weather API.

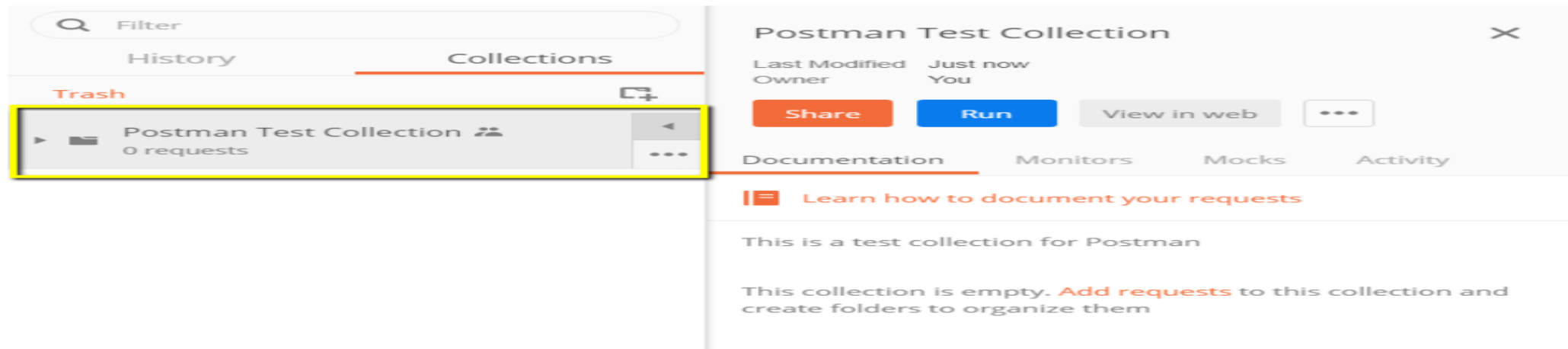
Collections in Postman- Creating Collections in Postman

- **Collections** are a group of requests grouped into one folder. So that we can use them and share them easily.
- Collections play an important role in organizing test suites. It can be imported and exported making it easy to share collections amongst the team.
- **Step 1)** Click on the New button at the top left corner of the page. Select Collection. Create collection window should pop up.

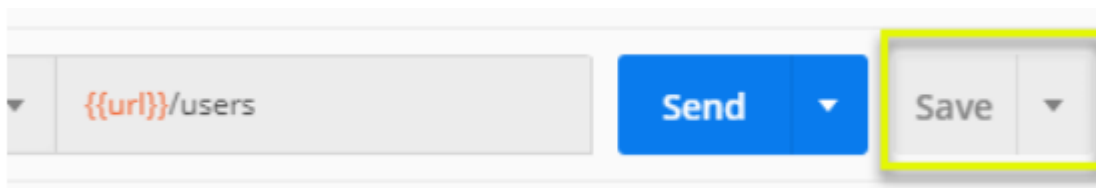


Collections in Postman- Creating Collections in Postman

- Step 2) Input the desired collection name and description then click create. A collection should now be created.



- Step 3) Go back to the previous Get request. Click Save



3.3: Environmental Variables in Postman

Collections in Postman- Creating Collections in Postman

- Step 4: Select Postman Test Collection.
- Click Save to Postman Test Collection

The screenshot shows the 'SAVE REQUEST' dialog box in Postman. The dialog has a dark header bar with the title 'SAVE REQUEST' and a close button. Below the header, there is a text area explaining that requests are saved in collections, with a link to 'Learn more about creating collections'. The 'Request name' field contains '{{url}}/users'. The 'Request description (Optional)' field contains the text 'Adding a description makes your docs better'. Below the description field, there is a section titled 'Select a collection or folder to save to:' which contains a search bar and a list of collections. The 'Postman Test Collection' is highlighted with a yellow box and a green circle with the number '1'. To the right of the list is a '+ Create Folder' button. At the bottom of the dialog, there are two buttons: 'Cancel' and 'Save to Postman Test Collection'. The 'Save to Postman Test Collection' button is highlighted with a yellow box and a green circle with the number '2'.

SAVE REQUEST

Requests in Postman are saved in collections (a group of requests).
[Learn more about creating collections](#)

Request name
{{url}}/users

Request description (Optional)
Adding a description makes your docs better

Descriptions support Markdown

Select a collection or folder to save to:

Search for a collection or folder

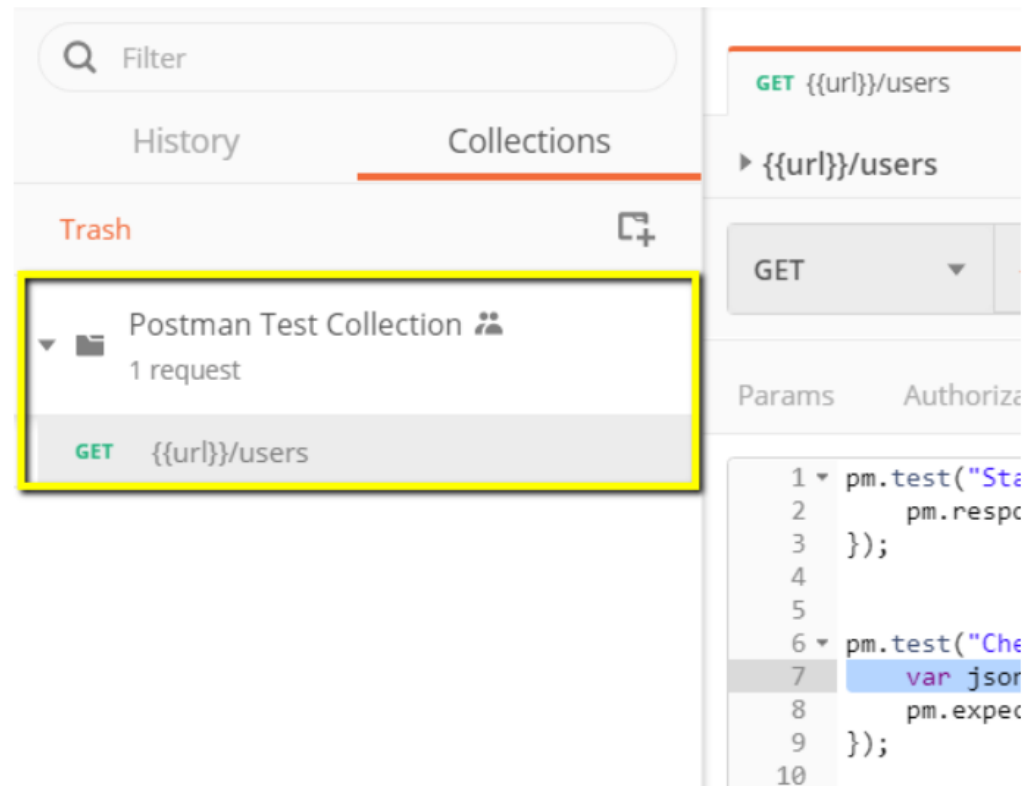
Postman Test Collection + Create Folder

Cancel Save to Postman Test Collection

3.3: Environmental Variables in Postman

Collections in Postman- Creating Collections in Postman

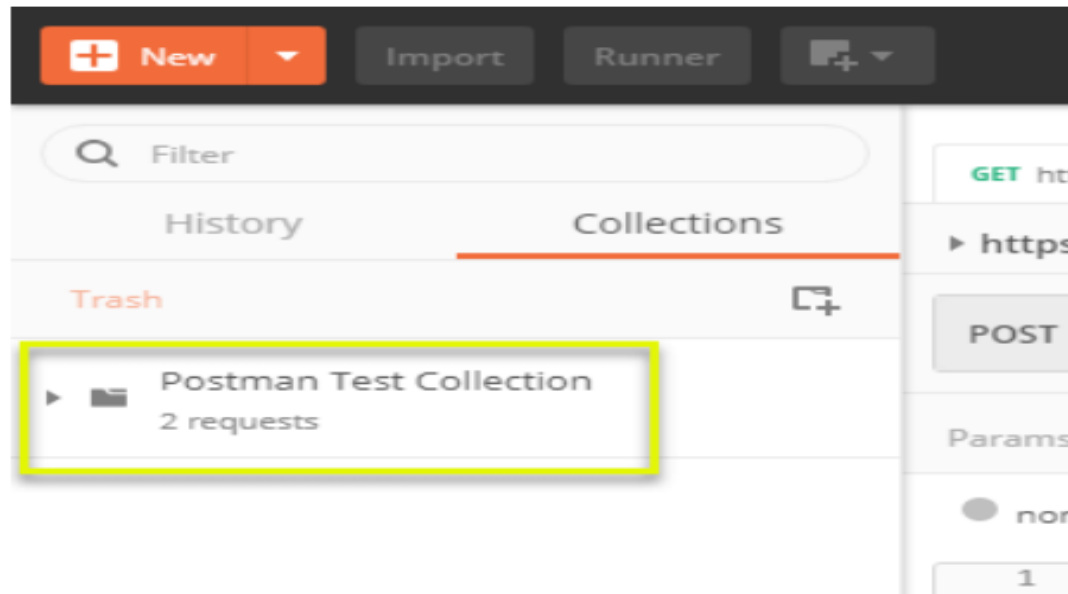
- Step 5) Postman test collection should now contain one request.



3.3: Environmental Variables in Postman

Collections in Postman- Creating Collections in Postman

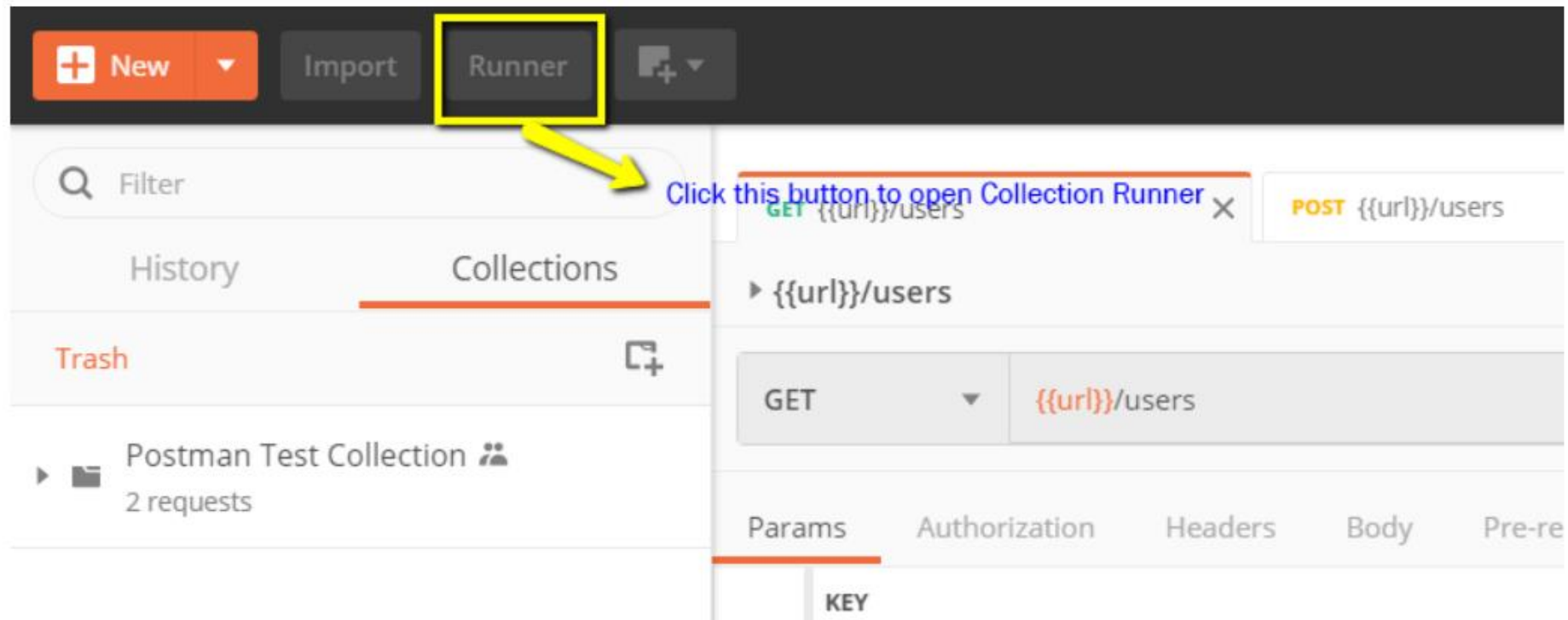
- Repeat steps 3-4 for the previous Post request so that collection will now have two requests.



3.3: Environmental Variables in Postman

Collections in Postman- Run Collections using Collection Runner

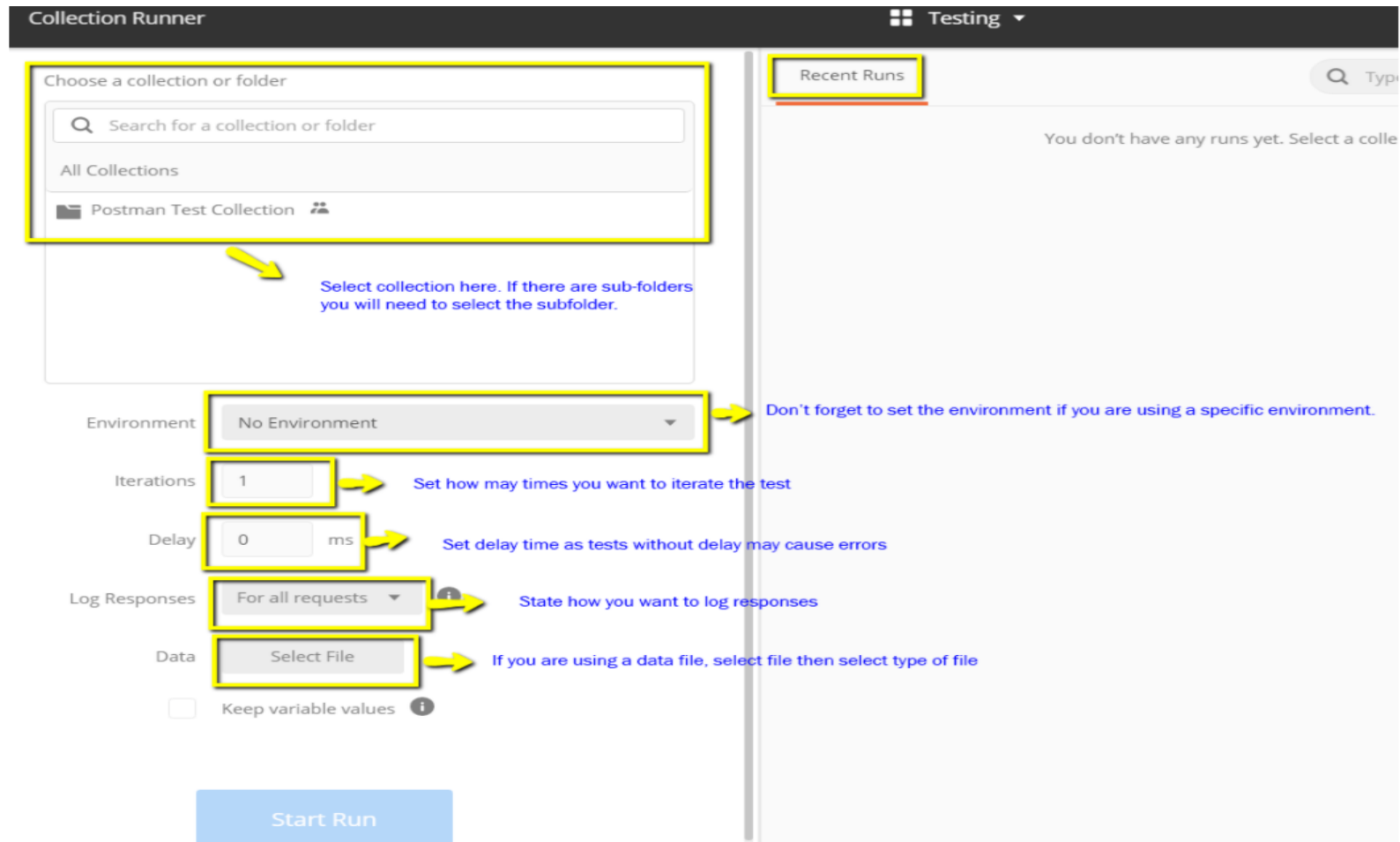
- Step 1): Click on the Runner button found at the top of the page next to the Import button.



3.3: Environmental Variables in Postman

Collections in Postman- Run Collections using Collection Runner

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

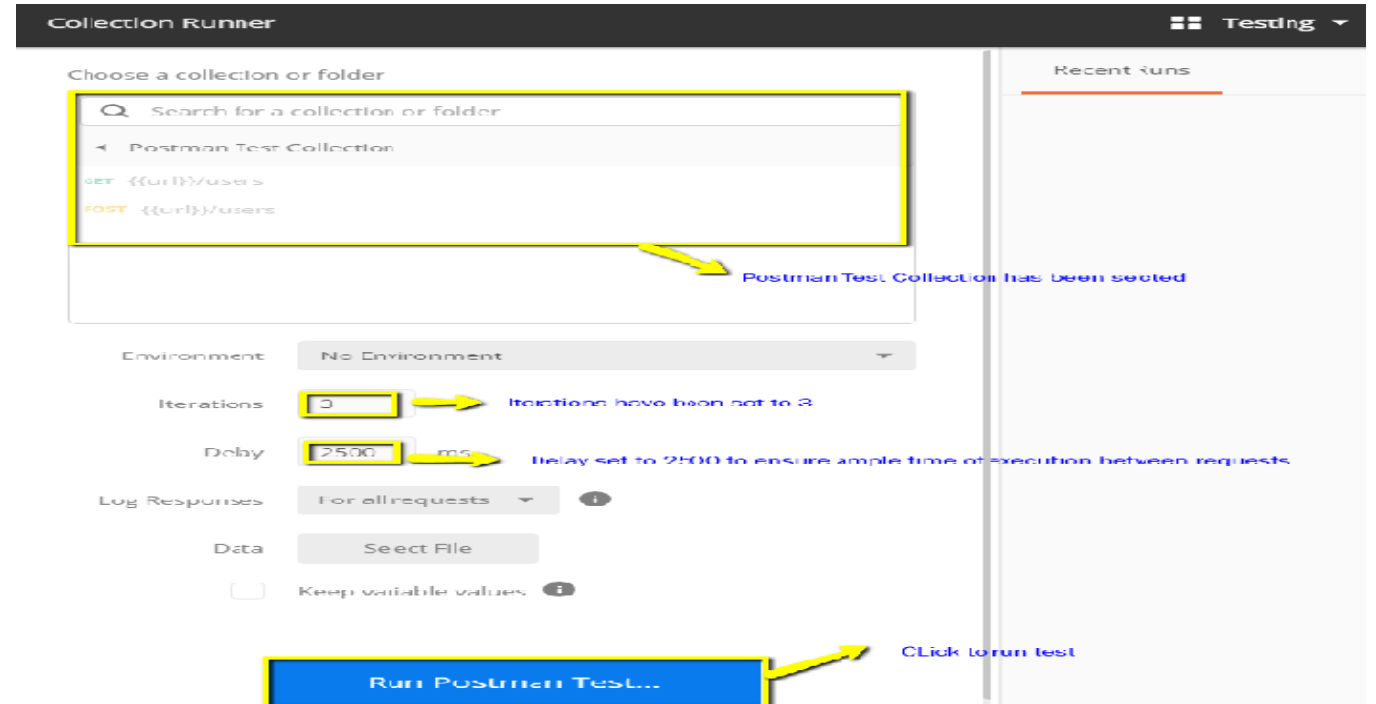


3.3: Environmental Variables in Postman

Collections in Postman- Run Collections using Collection Runner

■ 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



3.3: Environmental Variables in Postman

Collections in Postman- Run Collections using Collection Runner

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

The screenshot displays the 'Run Results' tab in Postman's Collection Runner. At the top, a summary shows 6 PASSED tests and 0 FAILED tests. Below this, the results are organized by iteration. Iteration 1 shows a GET request to 'https://jsonplaceholder.typicode.com/users' which passed with a status code of 200 and two assertions: 'Status code is 200' and 'Check if user with id1 is Leanne Graham'. The POST request to the same endpoint is shown below it with the message 'This request does not have any tests.' Iteration 2 and 3 show similar results for the GET request, while the POST request again shows no tests. Red callout numbers 1, 2, and 3 point to the summary, the GET test details, and the POST 'no tests' message respectively.

| Iteration | Request Method | URL | Status | Response | Time | Size |
|-------------|---------------------------------------|--|-------------|----------|--------|----------|
| Iteration 1 | GET | https://jsonplaceholder.typicode.com/users | 200 OK | 5.645 KB | 840 ms | 5.645 KB |
| | PASS | Status code is 200 | | | | |
| | PASS | Check if user with id1 is Leanne Graham | | | | |
| Iteration 1 | POST | https://jsonplaceholder.typicode.com/users | 201 Created | 487 B | 931 ms | 487 B |
| | This request does not have any tests. | | | | | |
| Iteration 2 | GET | https://jsonplaceholder.typicode.com/users | 200 OK | 5.645 KB | 79 ms | 5.645 KB |
| | PASS | Status code is 200 | | | | |
| | PASS | Check if user with id1 is Leanne Graham | | | | |
| Iteration 2 | POST | https://jsonplaceholder.typicode.com/users | 201 Created | 487 B | 823 ms | 487 B |
| | This request does not have any tests. | | | | | |
| Iteration 3 | | | | | | |

3.3: Environmental Variables in Postman

Assertions in Postman - What is assertion

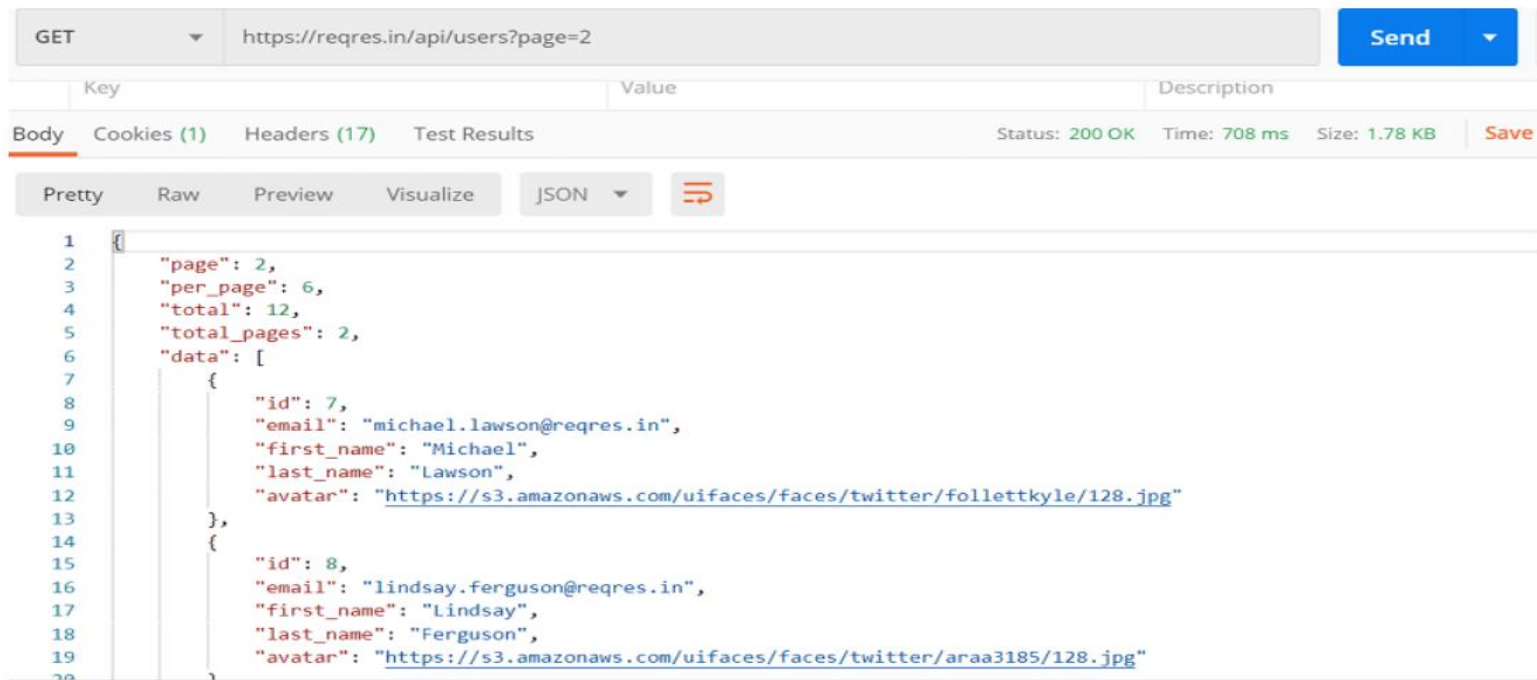
- **The assertion** checks whether the given predict is true or false
- A predict is a Boolean expression with the value true or false.
- assertion is nothing but code that can be used to check or analyze the response that is received from the server.
- Assertions help us to assert that expected and actual values should match during a test run.
- If the actual and expected values don't match, then the test will fail with the output pointing to the failure.

Assertions in Postman – How to use Assertion in Postman

- In Postman, we can write the assertion in many ways.
- One of the simplest ways is the snippets, which are nothing but a block of codes that have some unique functions inside it and are available in the postman app.
- In Postman, an assertion can be applied on different attributes such as an object, arrays, etc.
- By default in Postman, the received response from the server is displayed in the JSON format, we can select the JSON application and get the response converted into the JSON format.

3.3: Environmental Variables in Postman Assertions in Postman - Example

- enter the following URL in the URL text field : <https://reqres.in/api/users?page=2>
- Send the request and see the response:



3.3: Environmental Variables in Postman Assertions in Postman - Example

- Now add the following line of code in the test tab of the given request.

```
let jsonData = pm.response.json();  
console.log(jsonData.data);
```

GET ▼

https://reqres.in/api/users?page=2

Params ● Auth Headers (7) Body Pre-req. Tests ● Settings

```
1 let jsonData = pm.response.json();  
2 console.log(jsonData.data);
```

3.3: Environmental Variables in Postman Assertions in Postman - Example

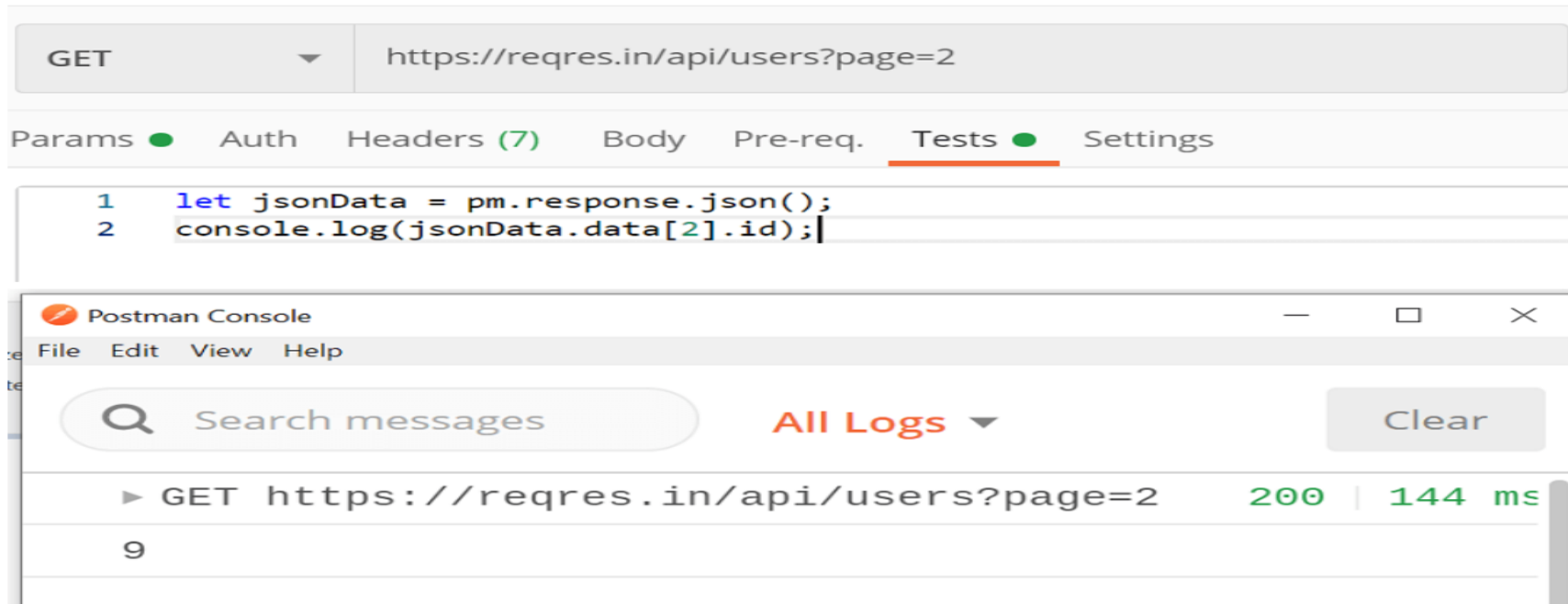
- Now, rerun the request and open the postman console.



3.3: Environmental Variables in Postman Assertions in Postman - Example

- Edit the tests editor with the following line of code and see the console:

```
let jsonData = pm.response.json();  
console.log(jsonData.data[2].id);
```



Summary

- In this lesson, you have learnt:
 - Basic Authentication in Postman
 - Environmental Variables in Postman
 - Collections in Postman
 - Assertions in Postman

