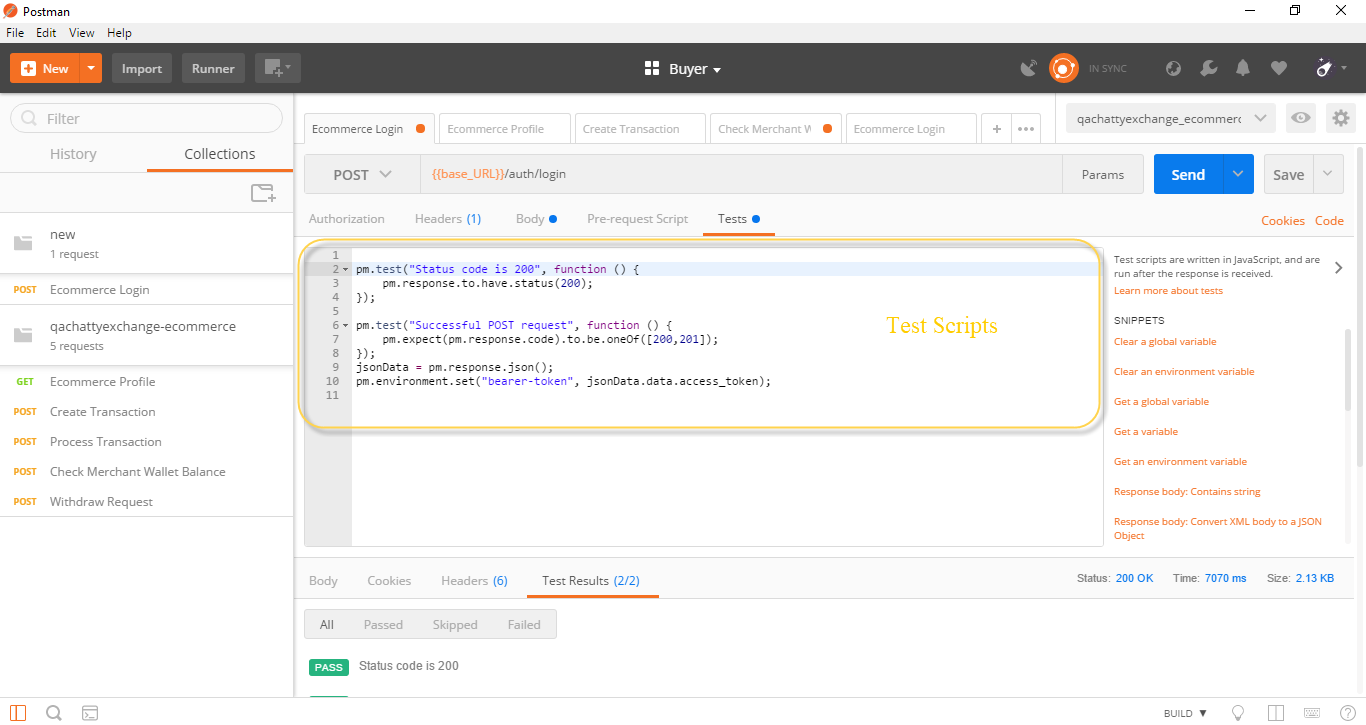
**TEST SCRIPTS**

1. **Basic Introduction to Test and Test Scripts**

A postman test is essentially JavaScript code executed after the request is sent, allowing to the pm.response object.

A postman test scripts in a block of JavaScript code which allows you to write and run the tests for each request.



1. **Accessing Data from Response Body**

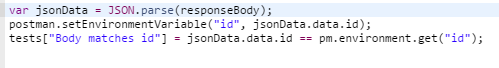
In the postman, we can access data from the response body to ensure that the test output is valid data or not and also to set and get variables to ease the further test.

The data in the response body may be in normal JSON or in array format. The JavaScript code written to access for normal format will not work to access the data in the array format. So here is the code how to access the both data format.

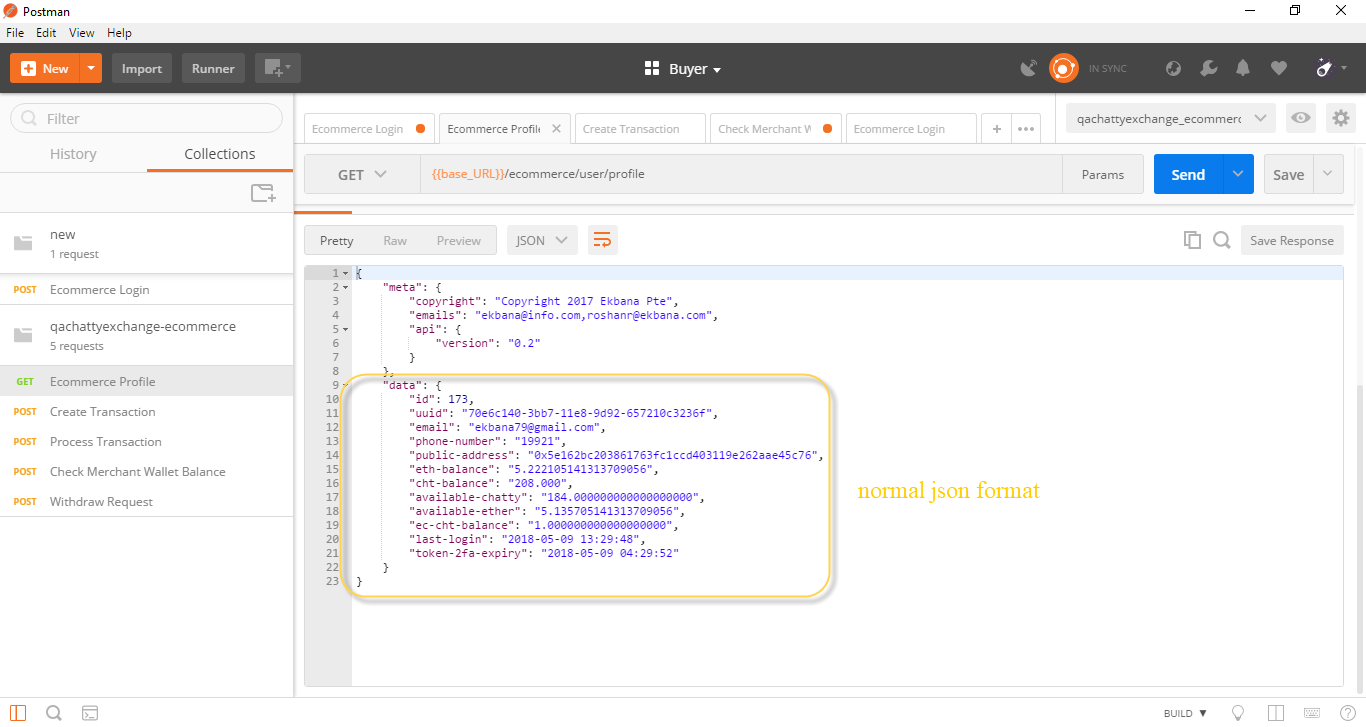
1. Normal JSON Format Data

Normal JSON format data can be seen enclosed in the {} braces.

How to access?



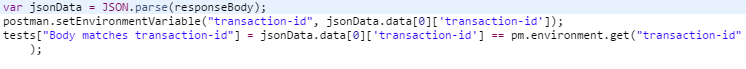
Here, we are trying to set the id into environment variable accessing data from response body. So the id is inside the data in response body in {} braces. So the code to access the id is jsonData.data.id (reference above figure).



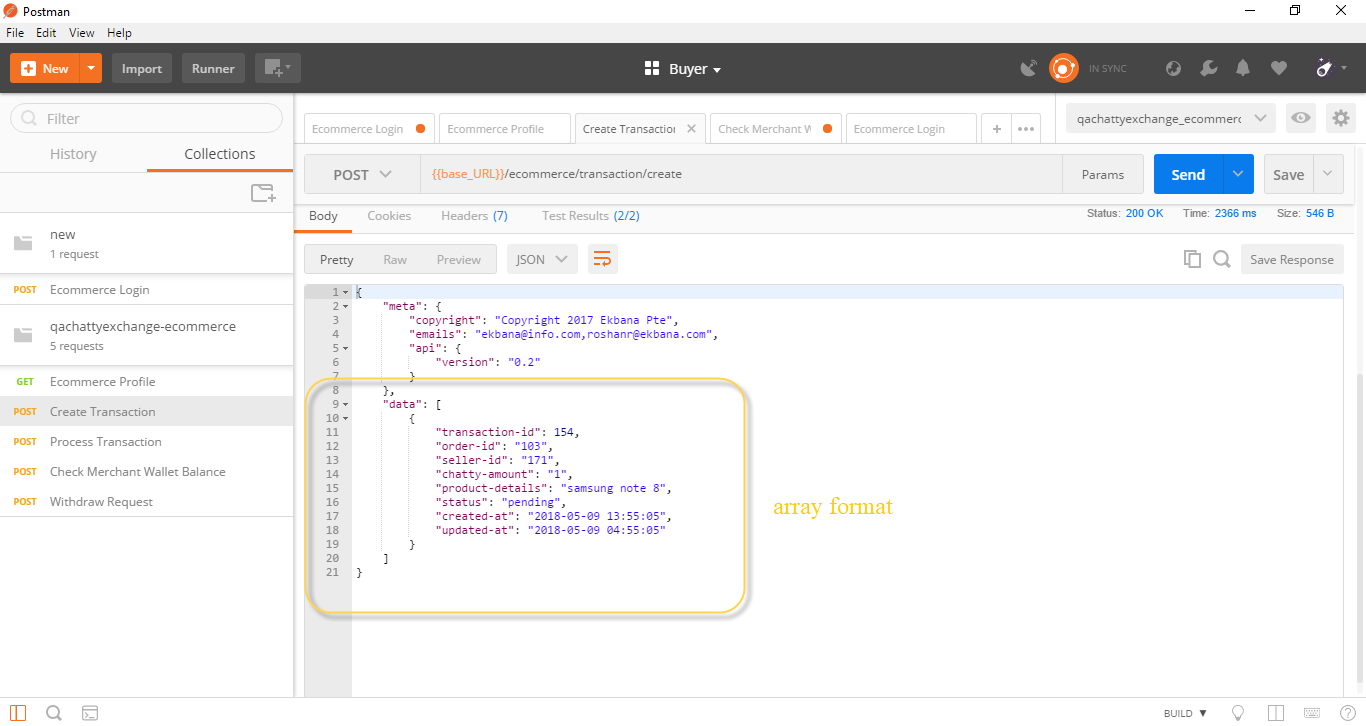
1. Array Format Data

Array format data can be seen enclosed in [ ] bracket.

How to access?



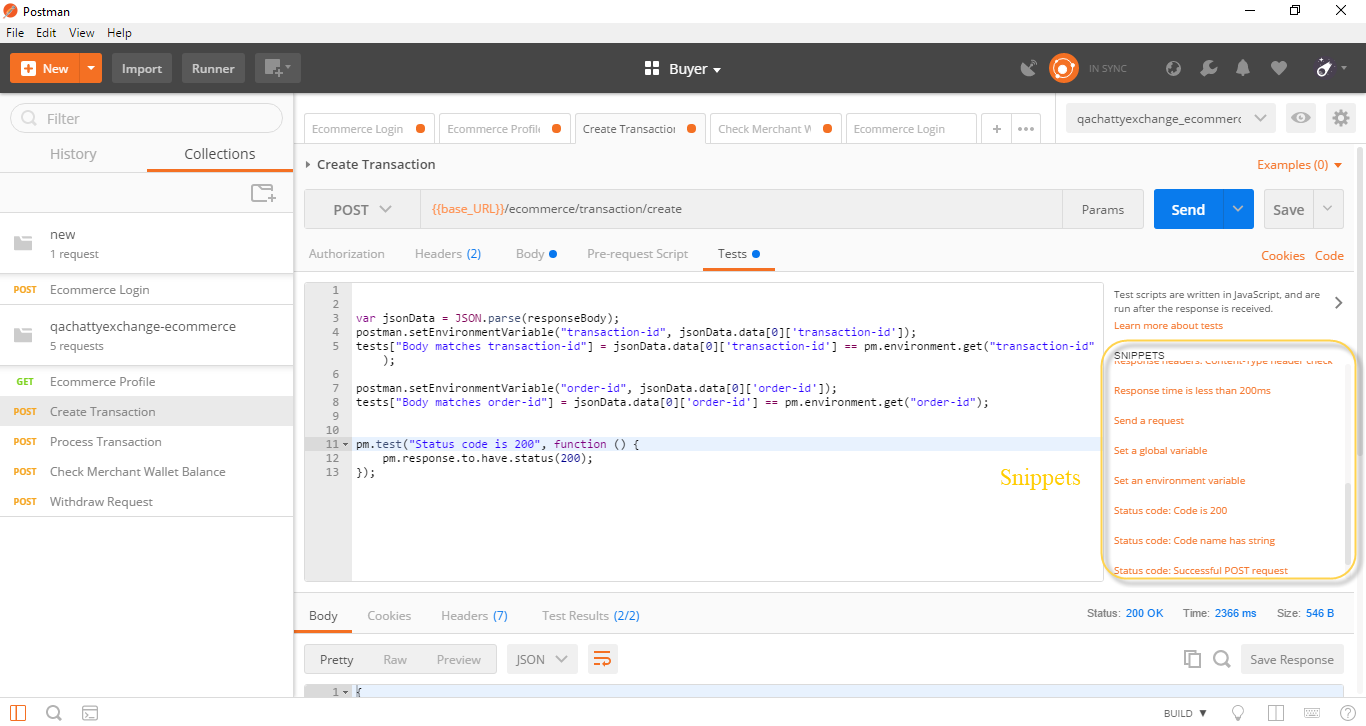
Here we are trying to set the transaction-id into environment variable accessing data from response body. So the transaction-id is inside the data in response body in [ ] bracket. So the code to access the transaction-id is jsonData.data[0][‘transaction-id’] (reference above figure). We here added [0] to access data due to the data here present is in array format.



1. **Snippets**

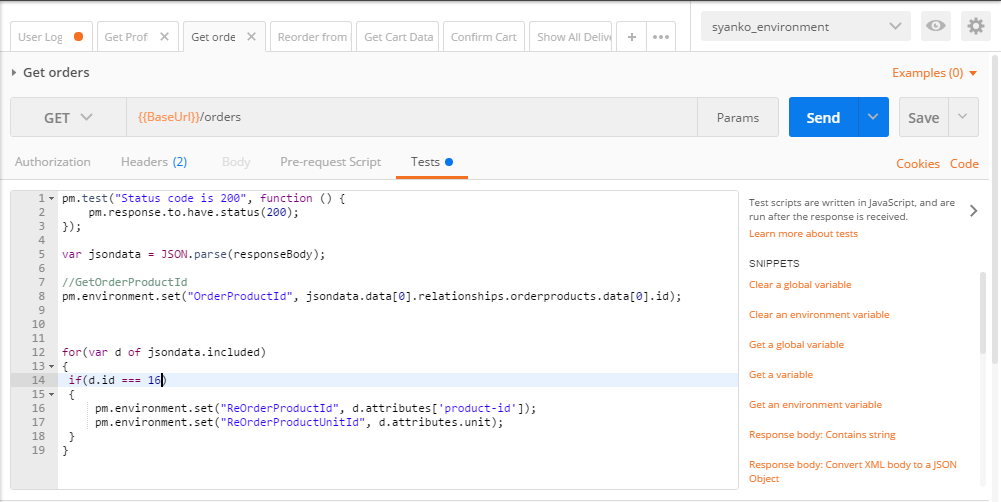
Snippets are the block of JS code which makes basic test easy and fast.

We can access the snippets block to check status code, set environment variable and so on.

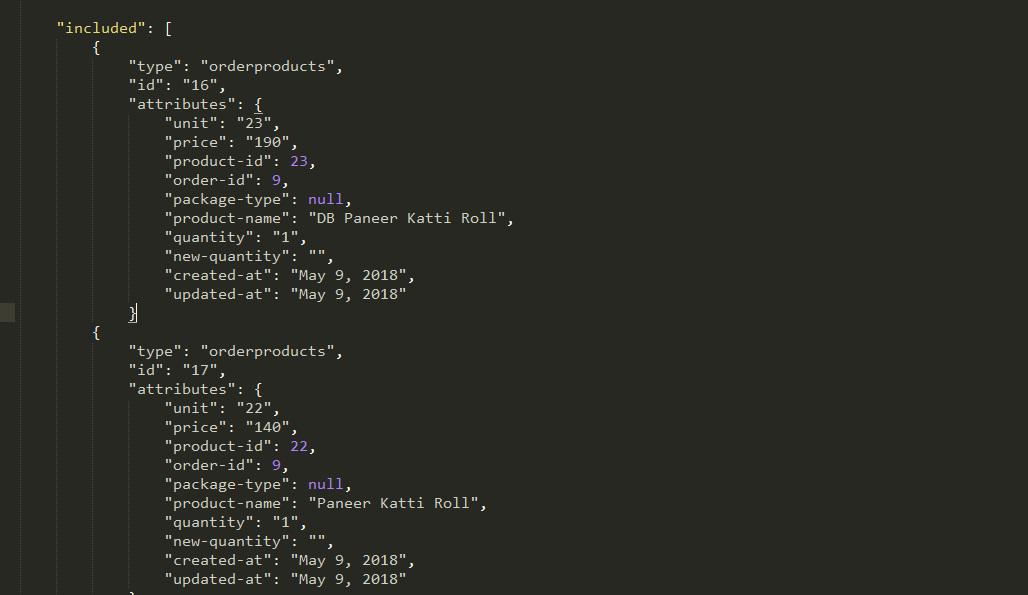


1. **Relating JS Code**

As per the API test in some request response might have multiple data in an array. To access this kind of data we must use loop and conditions in our test.



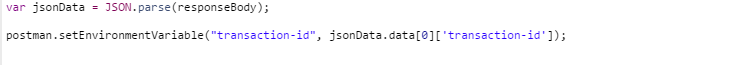
In the above figure, we have defined variable d which have certain value. The value here contains the attributes of the products. If the defined id matches the order number here it will set the attributes of that product in the environment variable.



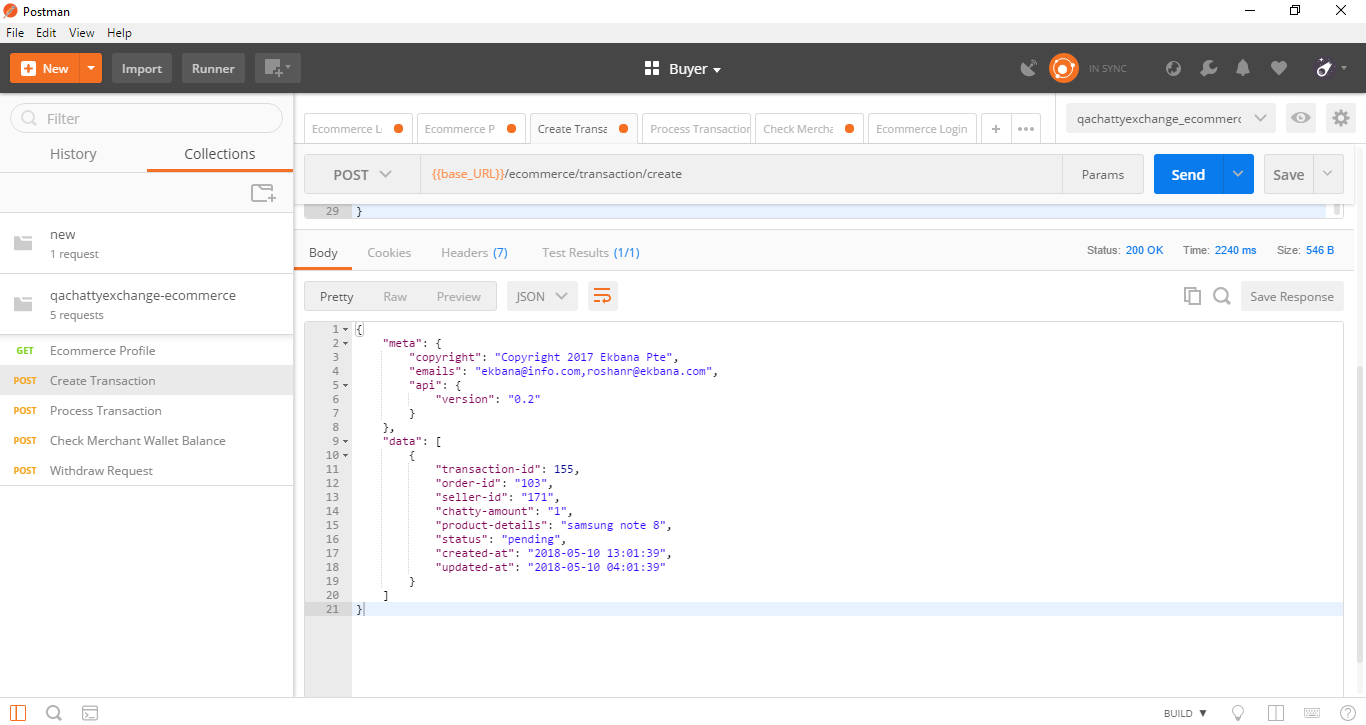
1. **SET and GET Variables**

Variables are symbols that can take different values. Most of the times variables are defined to reuse the values in multiple places so you can keep your code DRY (Don’t Repeat Yourself).

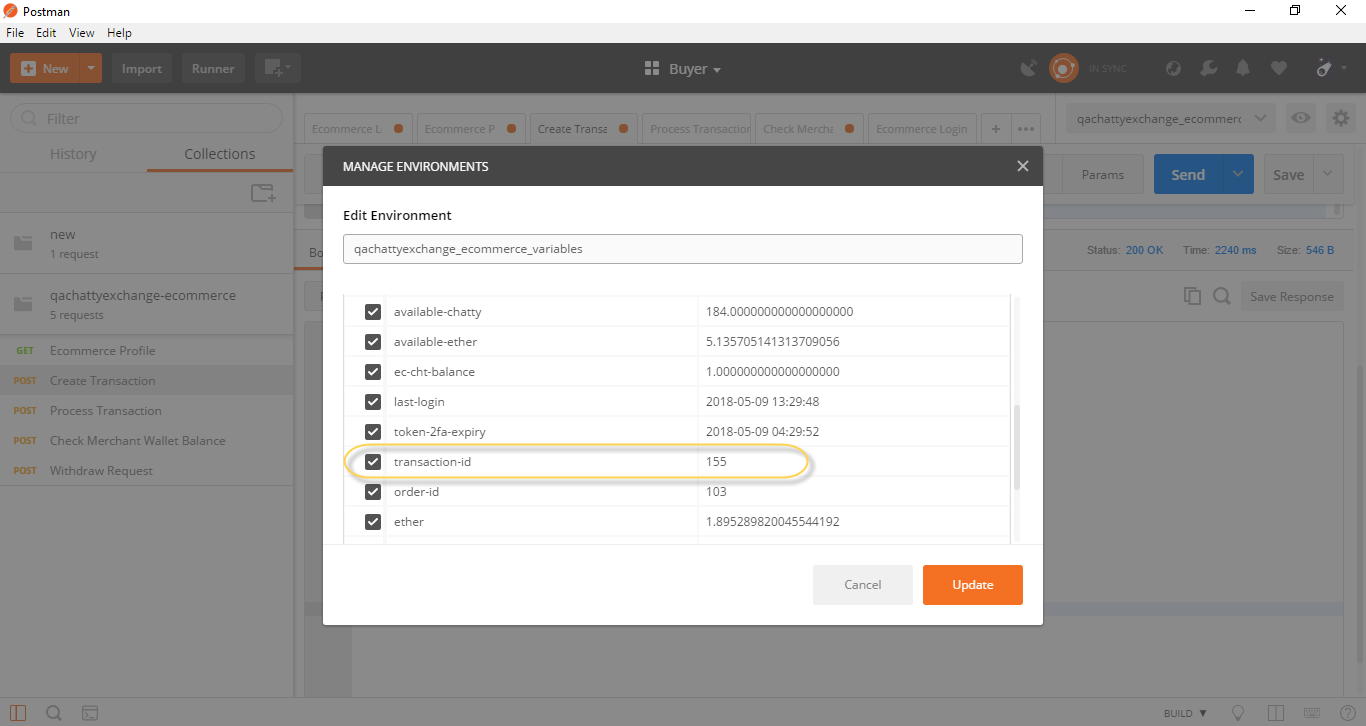
How to set?



The above code is use to set the environment variable.

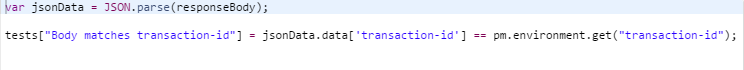


Here what we are trying to do is set transaction-id as an environment variable. So to set the transaction-id as environment variable we use the above code in the test scripts.



The transaction-id has been set as an environment variable. You can go to manage environments setting icon to view the variables that have been saved.

How to get?

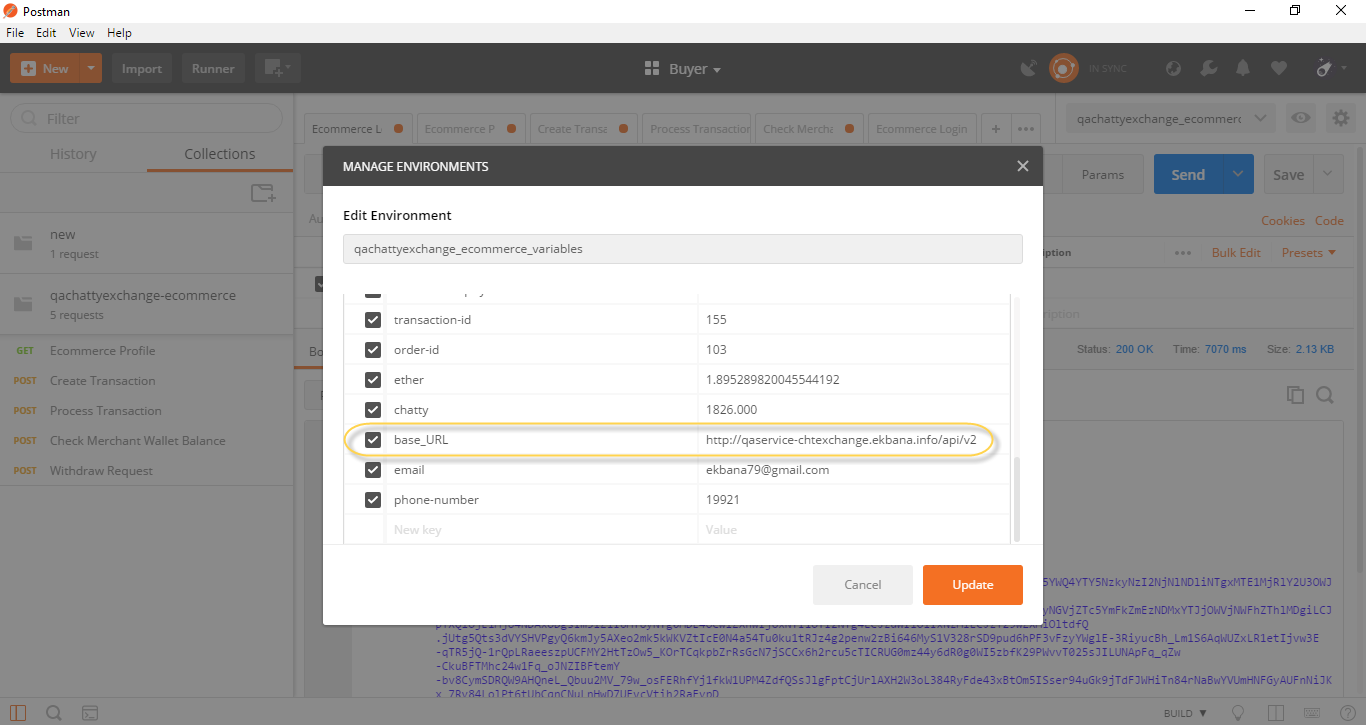


The above code is use to get the environment variable. The variable set is access through the get command. As per our project scenario we have to use transaction-id in multiple request so we set the transaction-id in environment variable and get the same transaction-id in other request which we have done.

1. **Passing/Using variables**

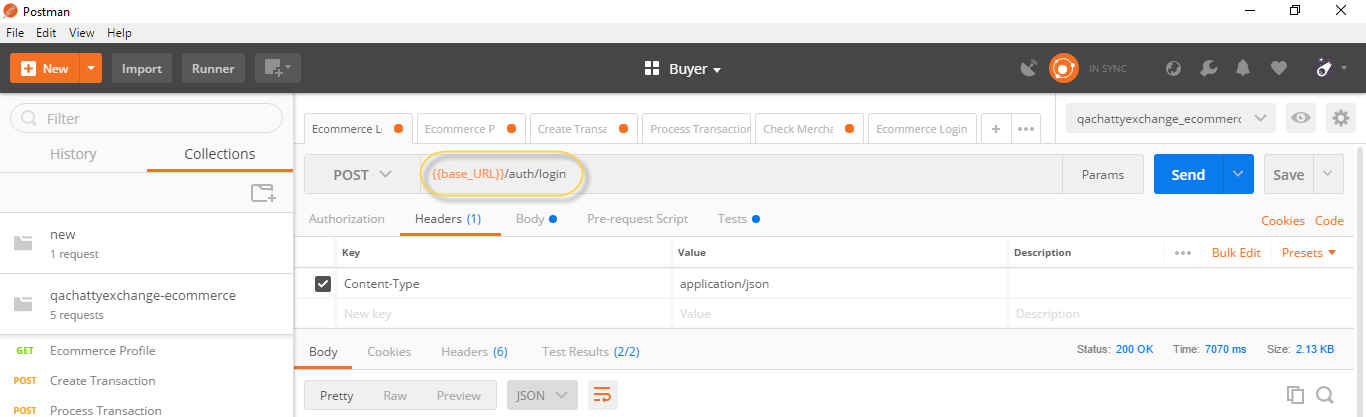
Variables can be used in the following form in the postman user interface- {{variableName}}.

The string {{variableName}} will be replaced with its corresponding value when postman resolves the variable.

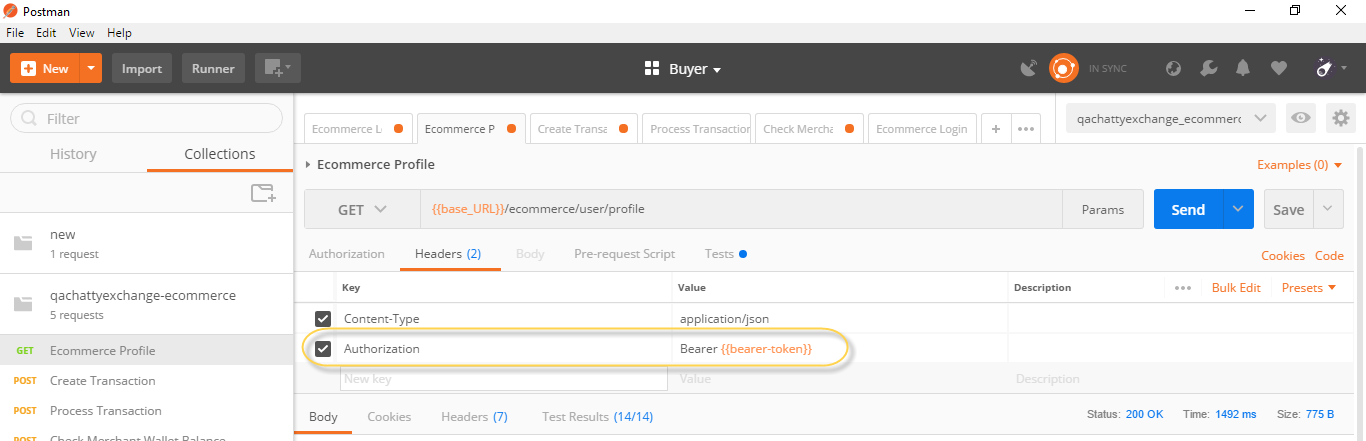


In the above figure, we have defined the base\_URL in the environment variable. Now we can access this variable in URL.

**i. Accessing the base\_URL variable in URL.**

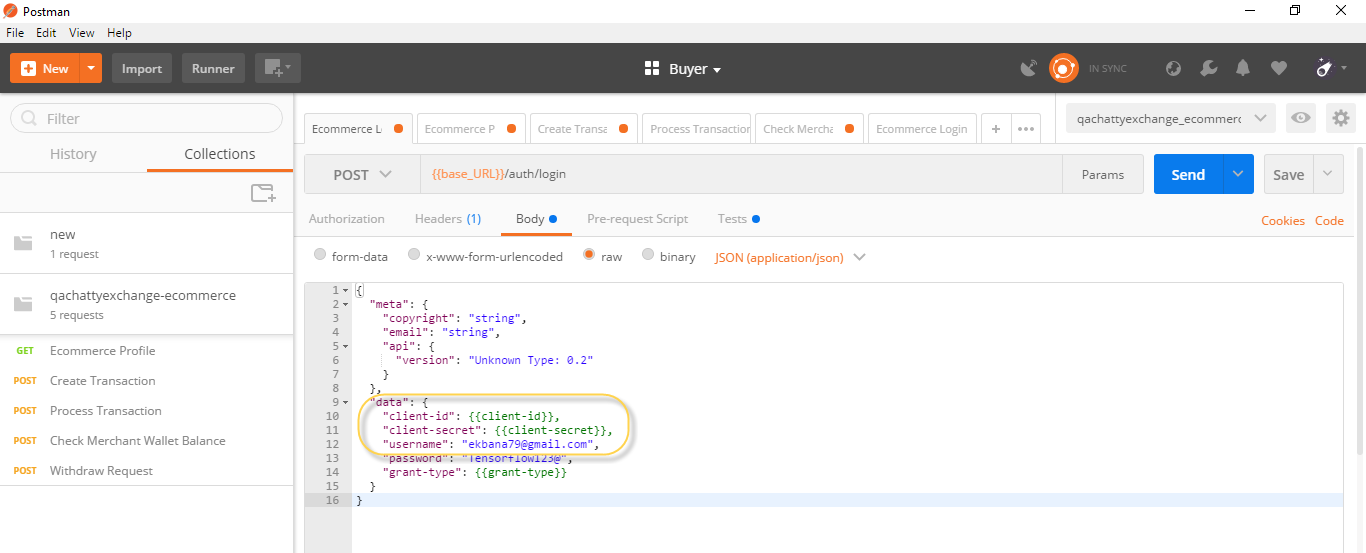


**ii. Accessing the Bearer token in Header**

****

As the base URL is defined in environment variable in the same way we defined Bearer token and pass the variable in the header as seen in the figure.

**iii. Accessing the variable in Request Body**

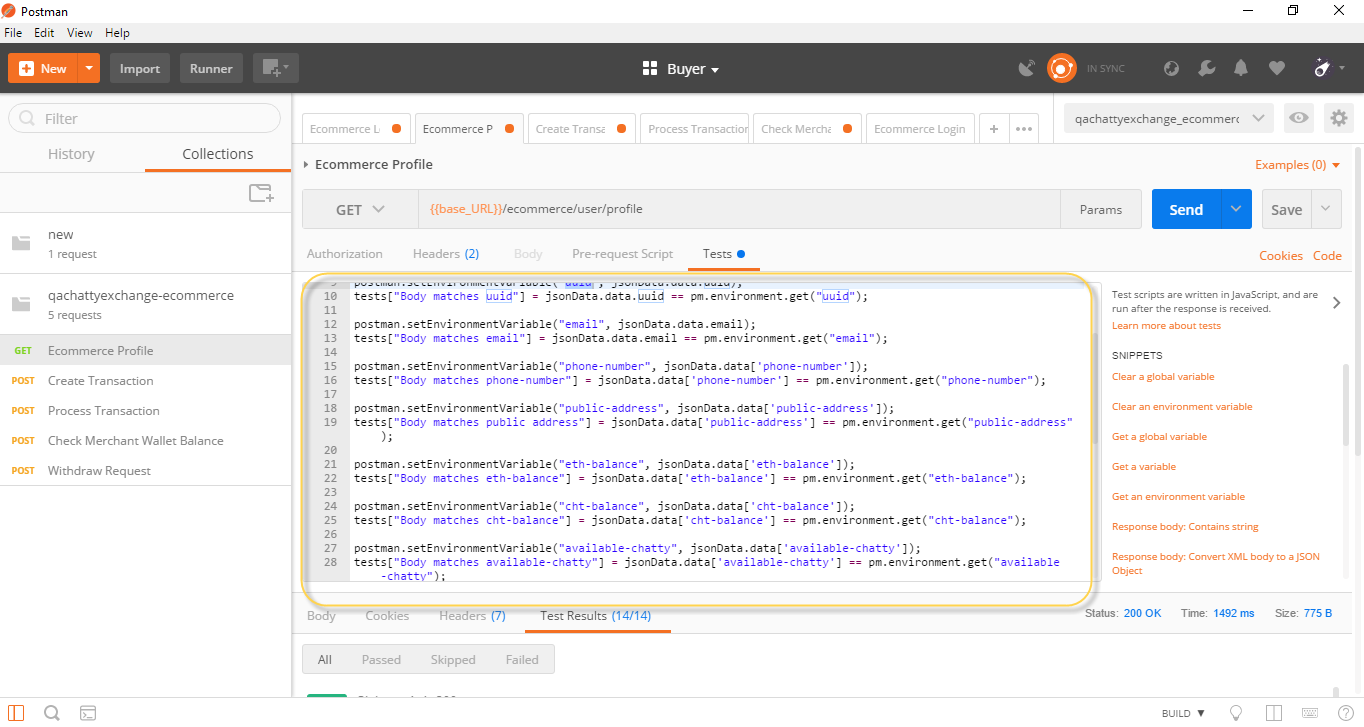
****

Here the client-id and client-secret id defined in the environment variable and pass the variable in the request body as {{client-id}} and {{client-secret}}.

In this way we can pass and use the variable in different areas.

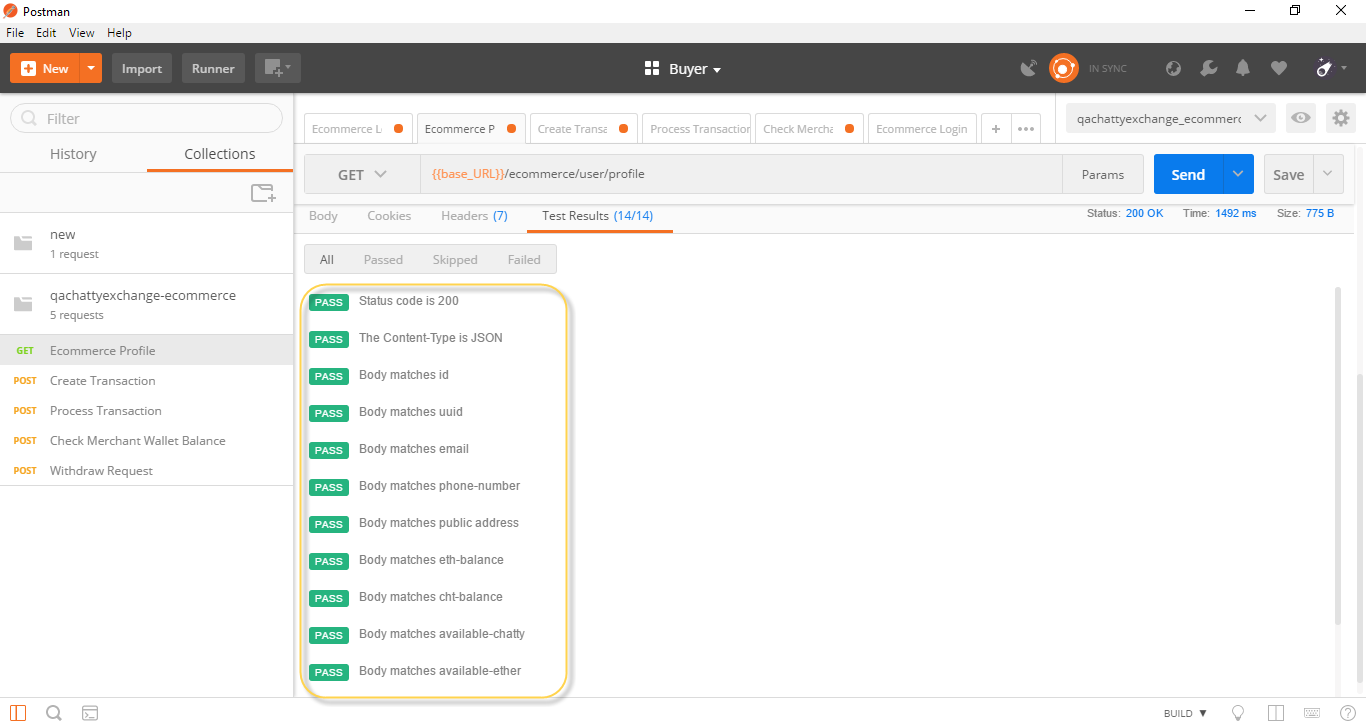
1. **Using Multiple Request in One Test**

The response body contains lots of data which you want to validate and check whether the response body data matches as per the requirement or not. So we have to test the response body is providing actual data or not. In this case we do multiple test to check the actual value are reflecting or not.



In the above figure we have used multiple test in one request to validate id, email, etc.

1. **Checking Test Response and Error Debugging**



The above figure shows the test results that we have tested and all the test have pass.

Not all the time test will pass, sometimes we mistype the name, we may be taking data in normal format but the data in response body may be in array format. So in that case we have to be careful and debug the errors.





In the above case, we have mistype the variable name. So the test has been failed due to assertion error.