# **Data Driven Testing in Postman**

***Last Updated On***

[**August 15, 2023**](https://qaautomation.expert/2023/08/09/data-driven-testing-in-postman/)

[**HOME**](https://qaautomation.expert/)

In this tutorial, we will see how we can use ***data files to feed test data to the Postman requests***.

Postman provides two ways to pass sets of values to use in a collection run.

1. JSON File
2. CSV File

By selecting a JSON or CSV data file in the Collection Runner, you can test your requests with multiple different values as part of a single run.

***Table of Contents***

1. [***Passing data using JSON File***](https://qaautomation.expert/2023/08/09/data-driven-testing-in-postman/#1-passing-data-using-json-file)
2. [***Create a request in Postman***](https://qaautomation.expert/2023/08/09/data-driven-testing-in-postman/#create-a-request-in-postman)
3. [***Create Tests for the request***](https://qaautomation.expert/2023/08/09/data-driven-testing-in-postman/#create-tests-for-the-request)
4. [***Run the Collection from Postman***](https://qaautomation.expert/2023/08/09/data-driven-testing-in-postman/#run-the-collection-from-postman)
5. [***Run the tests manually***](https://qaautomation.expert/2023/08/09/data-driven-testing-in-postman/#run-the-tests-manually)
6. [***Passing data using CSV File***](https://qaautomation.expert/2023/08/09/data-driven-testing-in-postman/#2-passing-data-using-csv-file)

### ***1. Passing data using JSON File***

Consider a sample JSON which has 4 sets of data. This data we will be passing to a POST request.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | [     {        "name":"Test1",        "job":"Developer"     },     {        "name":"Test2",        "job":"Scrum Master"     },     {        "name":"Test3",        "job":"Quality Analyst"     },     {        "name":"Test4",        "job":"Business Analyst"     }  ] |

#### ***Create a request in Postman***

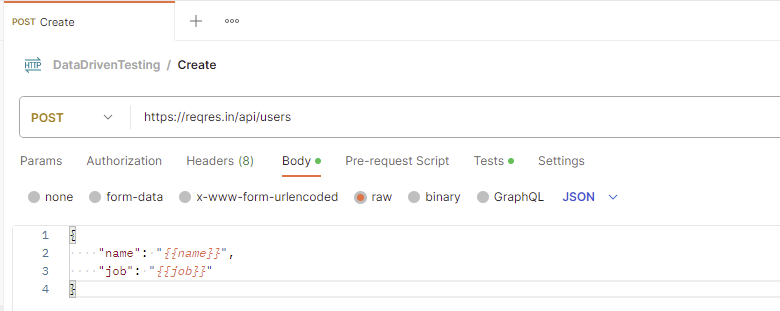
Create a request in Postman. Below is the URL of the request.

|  |  |
| --- | --- |
| 1 | https://reqres.in/api/users |

Paste the body of the request in the Body part of the Postman.

Advertisement

Here, name and job are the variables that we will be passing to the POST request. So, we have parameterized these 2 variables.

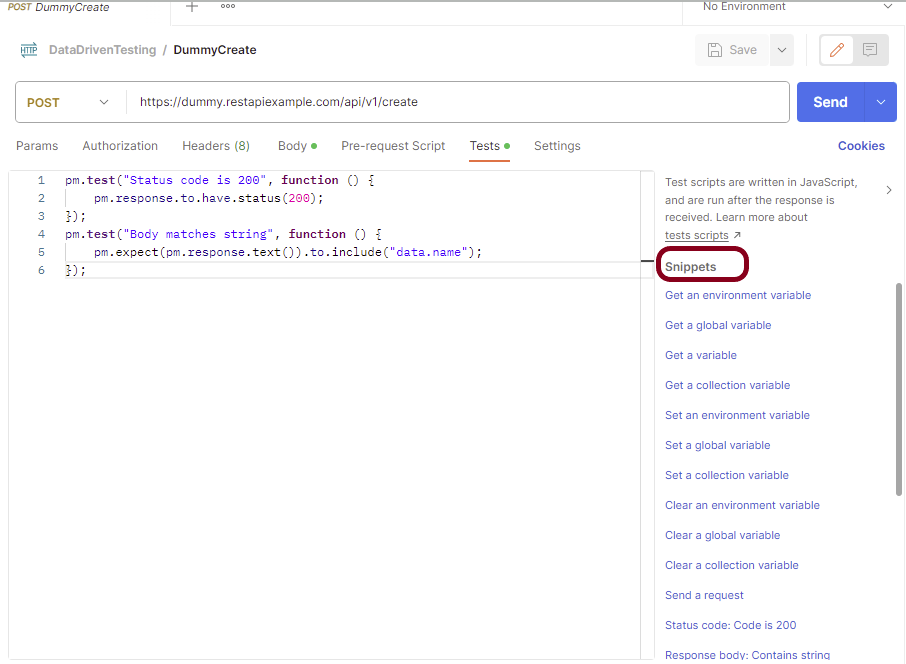


#### ***Create Tests for the request***

Click on the “Tests”. Write the tests as shown below:



If you need help in creating the test, you can refer to the snippets present on the right side of the console.

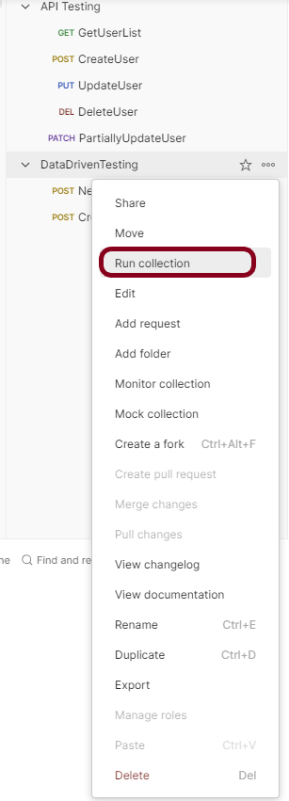


Here, data.json is the name of the json we are using in this example.

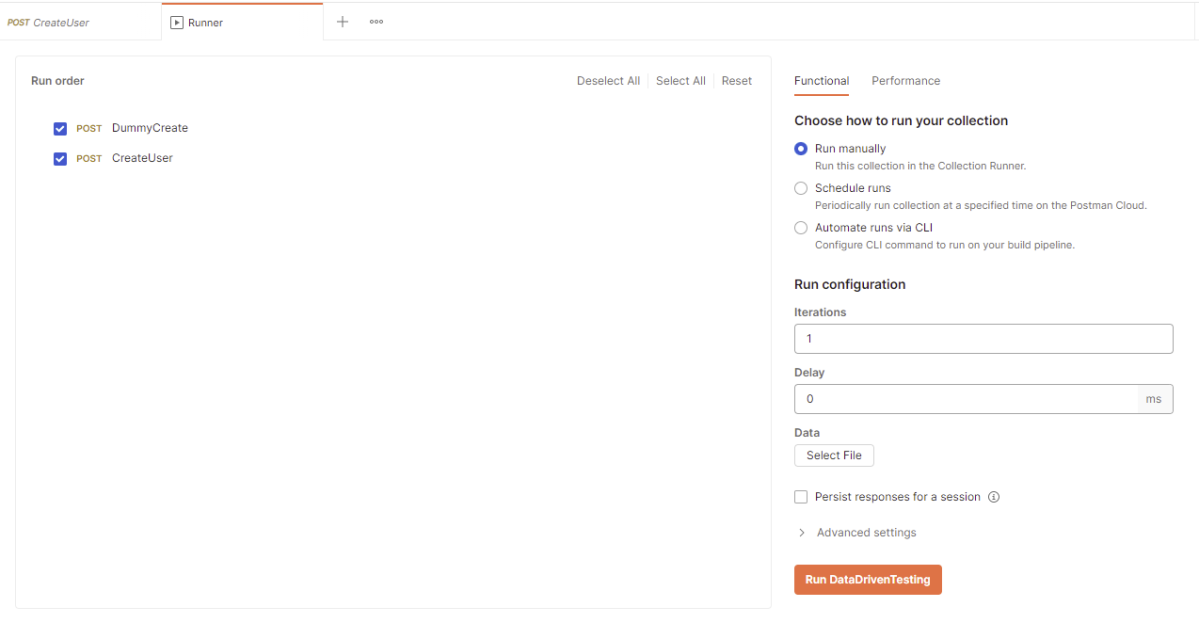
To read the JSON data we need to upload a JSON file from the collection runner.

#### ***Run the Collection from Postman***

Select the more actions icon  next to the collection name. Click on the ***“Run Collection”***option.

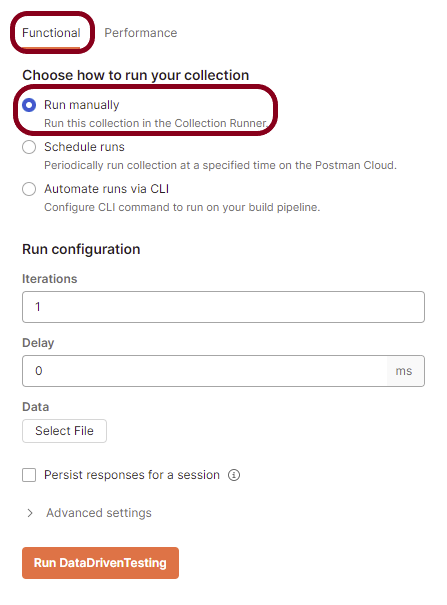


This shows all the requests present in this collection. Here, I want to run only ***CreateUser***Request. So, I will uncheck the***“Dummy User”***option.

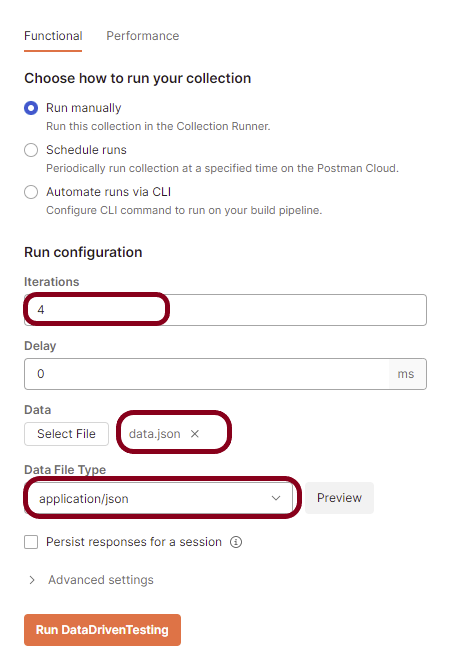


#### ***Run the tests manually***

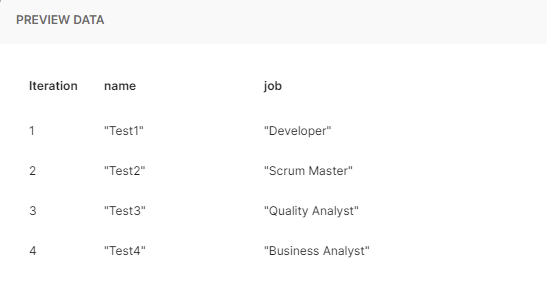
On the ***Functional***tab, select ***Run manually****.*



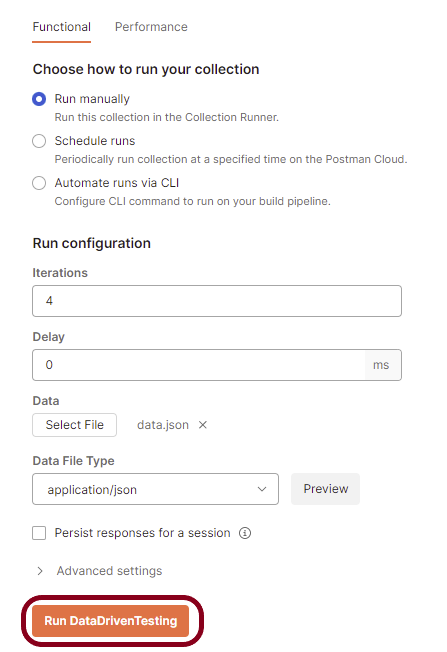
Click on the ***Select File*** and upload the data file. The Iterations field shows the number of data sets present in the data file. Here, we have 4 sets of data, so used Iterations as 4. The Delay field defines the time in ms between two iterations.



Click on the preview to see the data that is uploaded.



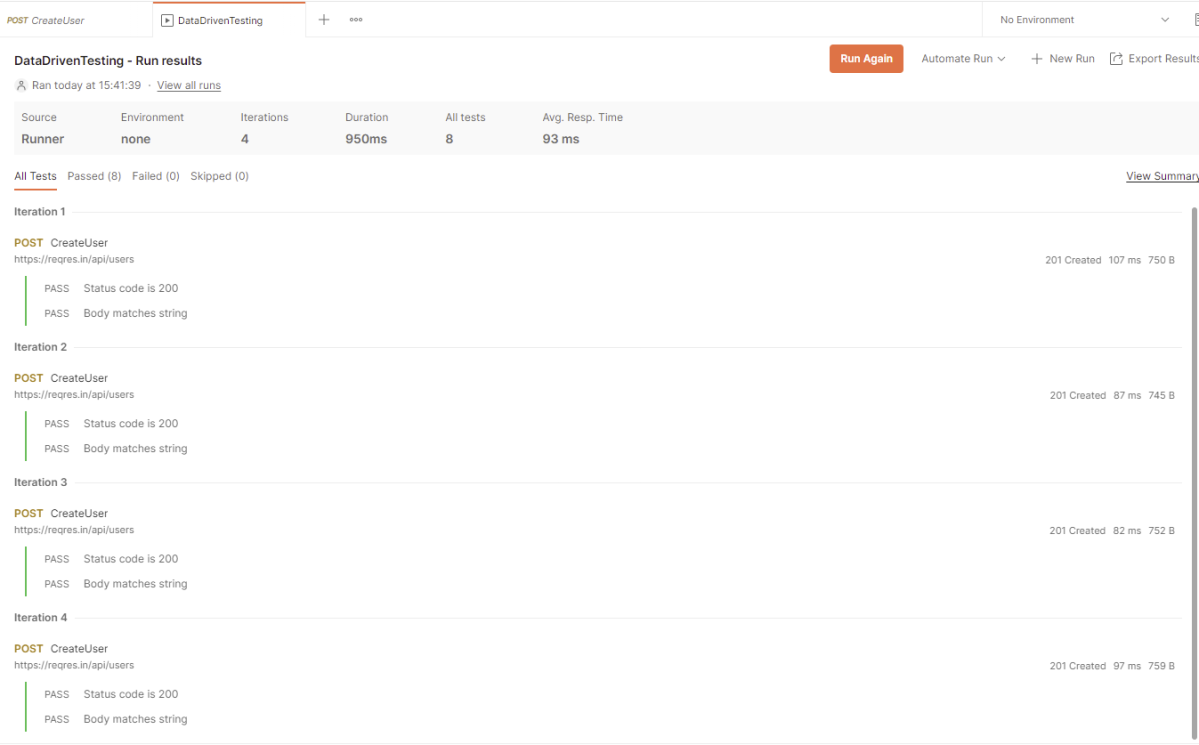
Click on the ***“RunDataDrivenTesting”*** button to run the tests available in the Collection.



Advertisements

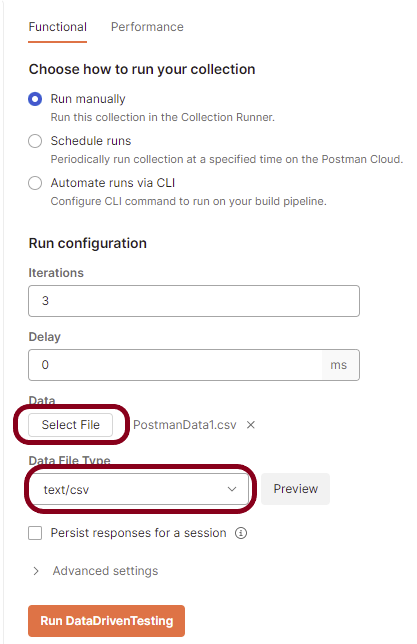
REPORT THIS AD

We can see that our API has run four times. Each time, a different set of data is picked up.

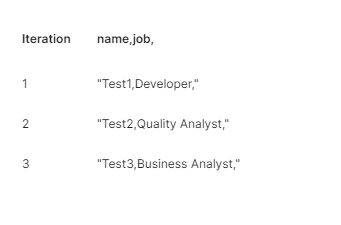


### ***2. Passing data using CSV File***

All the steps remain the same here, except instead of a JSON, we will use a .csv file.



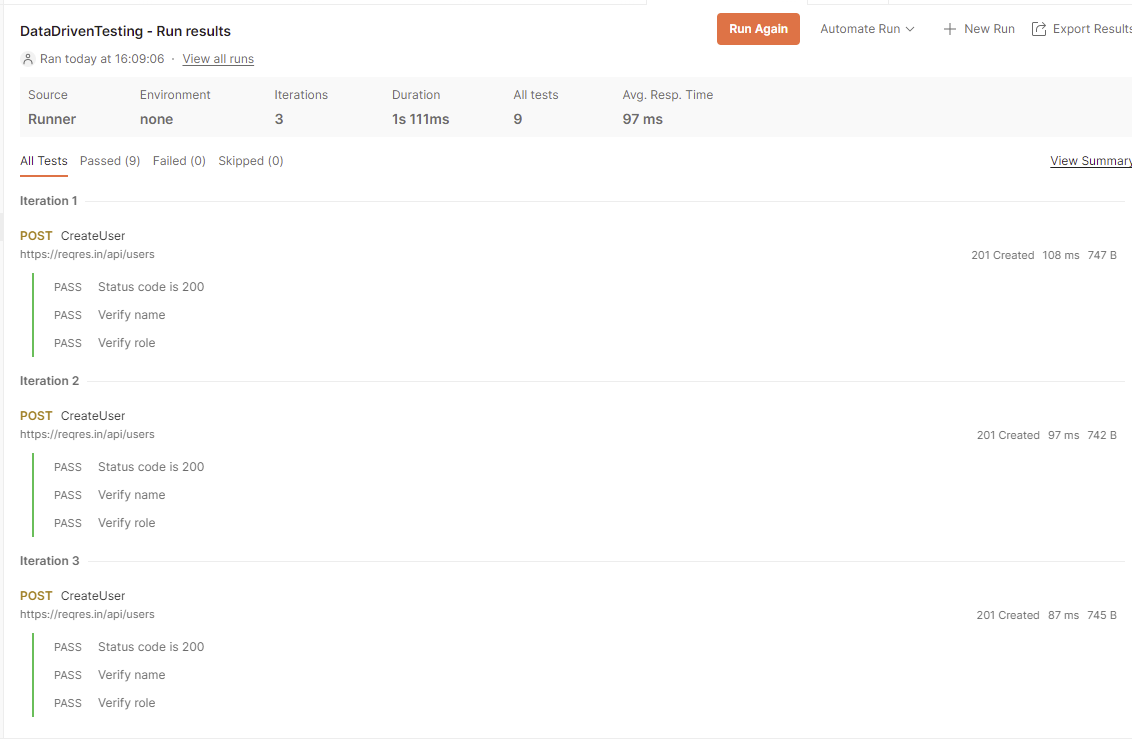
Click on the ***“Preview”*** button to view the data present in the .csv file.



Below are the sample tests.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | pm.test("Status code is 200", function () {      pm.response.to.have.status(201);  });    var expectedName = pm.variables.get("name");  var expectedRole = pm.variables.get("role");      pm.test("Verify name", function () {      var jsonData = pm.response.json();      pm.expect(jsonData.value).to.eql(expectedName);  });    pm.test("Verify role", function () {      var jsonData = pm.response.json();      pm.expect(jsonData.value).to.eql(expectedRole);  }); |

Run the collection again. Below is the image of the test execution in Postman.



Advertisements

REPORT THIS AD

We are done! Congratulations on making it through this tutorial and hope you found it useful! Happy Learning!!