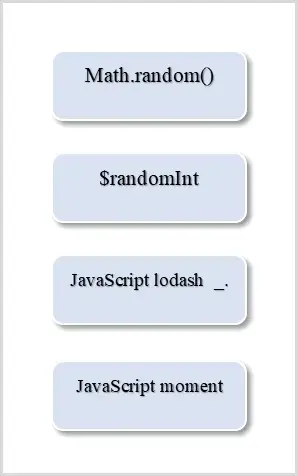
# Generate Random/Dynamic data in Postman Requests

In the previous articles on [Postman Tutorial](https://www.softwaretestingmaterial.com/postman-tutorial/), we have covered “[API Test Cases in Postman using JavaScript](https://www.softwaretestingmaterial.com/api-test-cases-in-postman-using-javascript/)“

In this “Generate Random/Dynamic Data in Requests” article, I will be demonstrating as to how you can implement this concept and get a tight grip over this.

We can generate random/dynamic data in requests using the following functions.

1. Math.random()
2. $randomInt
3. JavaScript Lodash \_.
4. JavaScript Moment



**Method 1: Using Math.random() function in Pre-request Scripts**

Suppose we want to generate a random number for **customerId** field every time an API is hit. Random function in the Math library is used to generate random numbers between 1 and 100. Then variable is set as a global variable and later on the global variable is used in the body to generate random data.

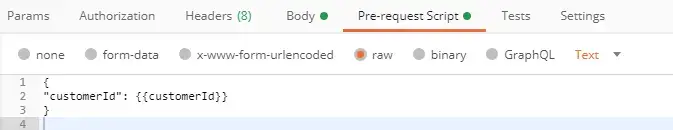
**Under Pre-request Scripts tab:**



const customerId = Math.floor((Math.random()\*100 +1);

pm.globals.set("customerId", customerId);

**Under Body tab:**



{

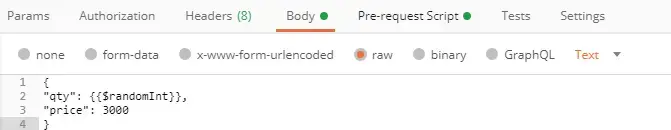
"customerId": {{customerId}},

}

**Method 2: Using Postman built-in variables to generate Random data**

Suppose we want to generate a random value for the quantity field. We can use Postman built-in variable **randomInt** to generate random integer value.

**Under Body tab:**



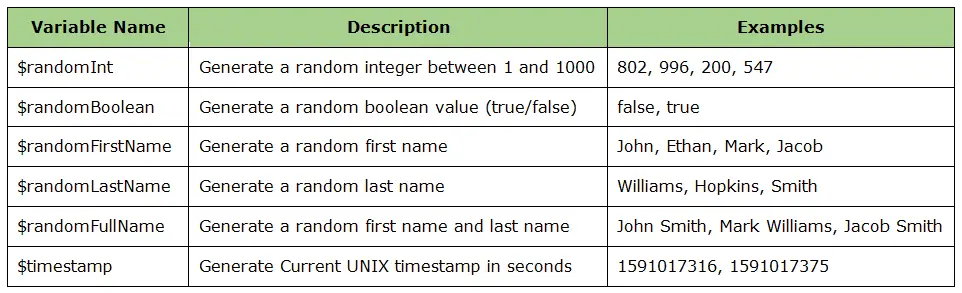
{

"qty": {{$randomInt}},

"price": 3000

}

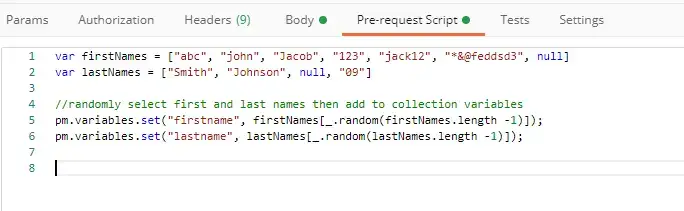
Other commonly used built-in variables are:



**Method 3: Using JavaScript Lodash module (\_.) to generate Random data**

We have some arrays of common first names and last names. Then we will be randomly picking things from those arrays and setting them in the Postman variables objects. The Postman variables are available to the collection while the tests are running, but then they go away so they’re not cluttering everything up while the tests aren’t running. Random functionality comes from a JavaScript library called **Lodash \_.** which is available automatically.

**Under Pre-request Scripts:**



var firstNames = ["abc", "john", "Jacob", "123", "jack12", "\*&@feddsd3", null]

var lastNames = ["Smith", "Johnson", null, "09"]

pm.variables.set("firstname", firstNames[\_.random(firstNames.length -1)]);

pm.variables.set("lastname", lastNames[\_.random(lastNames.length -1)]);

**Under Body tab:**



{

"firstname": pm.variables.get("firstname"),

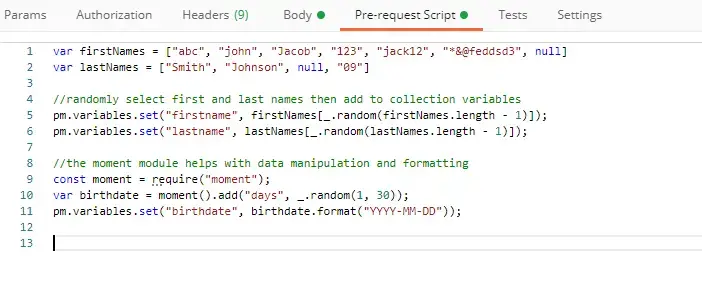
"lastname": pm.variables.get("lastname")

}

**Method 4: Using Moment module in JavaScript for date formatting**

Moment is a JavaScript library that parses, validates, manipulates, and displays dates and times in JavaScript. It makes it a lot easier than just the native JavaScript date. In order to input it and bring it into our script, we just need to do **require()** method. Once we have this **require()** method with moment, we can call this moment object to do all of our date manipulation.

**Under Pre-request Scripts:**



const moment = require("moment");

var birthdate = moment().add("days", \_.random(1,30));

pm.variables.set("birthdate", birthdate.format("YYYY-MM-DD"));

**Under Body tab:**



{

"birthdate": pm.variables.get("birthdate")

}