Title: Section 2: Postman tests

Proyect Name: Usa data api Test cases

Tester: Juan Pablo Navarro Quintero

Test Type: Postman Api requests

Activity Date: 14/7/2024

GET request with the api direction given: https://datausa.io/api/data?drilldowns=Nation&measures=Population&year={{ano\_actual}}

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | ID | Module | Test case Name | Description | Preconditions | Steps | Expected Outcome | Obtained Outcome | Status |
| 1 | PT1 | GET | PT1\_GET\_validate\_200\_status\_code | Validate a correct answer and receiving a 200 status code | 1. Add a year parameter, which gets it’s value from a global variable created in the pre-request Script, to the GET instruction  2. There must be a properly configured HTTP request pointing to the endpoint you want to test. | 1. Insert the following pre-request script (it creates a global variable used as parameter):  **var** moment = require('moment');  pm.globals.**set**('ano\_actual', moment().format(("YYYY")) -2);  2. Insert a script test that validates the 200 status code:  pm.test("El codigo del estatus es 200", **function** () {      pm.response.to.have.status(200);  });  3. Execute the GET request with the api direction given. | The HTTP status code returned by the  response should be 200, indicating that  the request was successful. | The HTTP status code returned by the response should be 200, indicating that the request was successful. | **Pass** |
| 2 | PT2 | GET | PT2\_GET\_validate\_request\_time | The response time should be less than 800 milliseconds (or 5 seconds), which indicates a timely response from the server. | 1. Add a year parameter, which gets it’s value from a global variable created in the pre-request Script, to the GET instruction  2. There must be a properly configured HTTP request pointing to the endpoint you want to test. | 1. Insert the following pre-request script (it creates a global variable used as parameter):  **var** moment = require('moment');  pm.globals.**set**('ano\_actual', moment().format(("YYYY")) -2);  2. Insert a script test that validates the responds time:  pm.test("El tiempo de respuesta es menor que 800 ms", function () {  pm.expect(pm.response.responseTime).to.be.below(800);});  3. Execute the GET request with the api direction given. | Receive a value indicating that the respond time is less than 800 ms. | Received a value of 416 ms of respond time. | **Pass** |
| 3 | PT3 | GET | PT3\_GET\_validate\_Content\_  type\_header | Validate that the obtained header data contains the “Content-type”. | 1. Add a year parameter, which gets it’s value from a global variable created in the pre-request Script, to the GET instruction  2. There must be a properly configured HTTP request pointing to the endpoint you want to test. | 1. Insert the following pre-request script (it creates a global variable used as parameter):  **var** moment = require('moment');  pm.globals.**set**('ano\_actual', moment().format(("YYYY")) -2);  2. Insert the following script test:  pm.test("Content-Type esta presente en el encabezado", **function** () {      pm.response.to.have.header("Content-Type");  });  3. Execute the GET request with the api direction given. | The header Content-type is found | The header Content-type is found | **Pass** |
| 4 | PT4 | GET | PT4\_GET\_validate\_not\_null\_value\_  Nation\_variable\_of\_first\_vector\_  column | Validate that the first Nation variable of the first column of the obtained vector is not null | 1. Add a year parameter, which gets it’s value from a global variable created in the pre-request Script, to the GET instruction  2. There must be a properly configured HTTP request pointing to the endpoint you want to test. | 1. Insert the following pre-request script (it creates a global variable used as parameter):  **var** moment = require('moment');  pm.globals.**set**('ano\_actual', moment().format(("YYYY")) -2);  2. Insert the following script test:  pm.test("Valida que en el primer campo del vector data, la variable 'Nation' no es null", **function** () {  **var** jsonData = pm.response.json();      pm.expect(jsonData.data[0].Nation).is.not.**null**;  });  3. Execute the GET request with the api direction given. | The test confirms that the Nation variable is not null. | The test confirms that the Nation variable is not null. | **Pass** |
| 5 | PT5 | GET | PT5\_GET\_validate\_value\_  Nation\_variable\_of\_first\_vector\_  Column\_is\_United\_States | Validate that the first Nation variable of the first column of the obtained vector is equals to “United States” | 1. Add a year parameter, which gets it’s value from a global variable created in the pre-request Script, to the GET instruction  2. There must be a properly configured HTTP request pointing to the endpoint you want to test. | 1. Insert the following pre-request script (it creates a global variable used as parameter):  **var** moment = require('moment');  pm.globals.**set**('ano\_actual', moment().format(("YYYY")) -2);  2. Insert the following script test:  pm.test("Valida que la variable Nation sea igual a 'United States'", **function** () {  **var** jsonData = pm.response.json();      pm.expect(jsonData.data[0].Nation).to.eql("United States");  });  3. Execute the GET request with the api direction given. | The test confirms that the Nation variable is equals to “United States” | The test confirms that the Nation variable is equals to “United States” | **Pass** |
| 6 | PT6 | GET | PT6\_GET\_validate\_value\_  Nation\_variable\_of\_first\_vector\_  Column\_contains\_string\_United | Validate that the first Nation variable of the first column of the obtained vector contains the string “United” | 1. Add a year parameter, which gets it’s value from a global variable created in the pre-request Script, to the GET instruction  2. There must be a properly configured HTTP request pointing to the endpoint you want to test. | 1. Insert the following pre-request script (it creates a global variable used as parameter):  **var** moment = require('moment');  pm.globals.**set**('ano\_actual', moment().format(("YYYY")) -2);  2. Insert the following script test:  pm.test("Valida que la variable Nation contenga el string 'United'", **function** () {  **var** jsonData = pm.response.json();      pm.expect(jsonData.data[0].Nation).to.contain("United");  });  3. Execute the GET request with the api direction given. | The test confirms that the Nation variable contains the string “United” | The test confirms that the Nation variable contains the string “United” | **Pass** |
| 7 | PT7 | GET | PT7\_GET\_validate\_data\_type\_  Nation\_variable\_of\_first\_vector\_  Column\_is\_string | Validate that the first Nation variable of the first column of the obtained vector is a string data type | 1. Add a year parameter, which gets it’s value from a global variable created in the pre-request Script, to the GET instruction  2. There must be a properly configured HTTP request pointing to the endpoint you want to test. | 1. Insert the following pre-request script (it creates a global variable used as parameter):  **var** moment = require('moment');  pm.globals.**set**('ano\_actual', moment().format(("YYYY")) -2);  2. Insert the following script test:  pm.test("Valida que la variable Nation sea tipo string", **function** () {  **var** jsonData = pm.response.json();      pm.expect(jsonData.data[0].Nation).to.be.a("string");  });  3. Execute the GET request with the api direction given. | The test confirms that the Nation variable is a string data type | The test confirms that the Nation variable is a string data type | **Pass** |
| 8 | PT8 | GET | PT8\_GET\_validate\_year\_obtained\_is  \_equals\_global\_variable | Validate that the ‘year’ variable obtained is equals to the ‘ano\_actual’ global variable | 1. Add a year parameter, which gets it’s value from a global variable created in the pre-request Script, to the GET instruction  2. There must be a properly configured HTTP request pointing to the endpoint you want to test. | 1. Insert the following pre-request script (it creates a global variable used as parameter):  **var** moment = require('moment');  pm.globals.**set**('ano\_actual', moment().format(("YYYY")) -2);  2. Insert the following script test:  pm.test("valida la igualdad de la  variable 'year' del objeto filtrado contra la variable global utilizada en los parámetros", **function** () {    pm.expect(pm.response.json().year).to.eql(pm.environment.**get**("ano\_actual"));  });  3. Execute the GET request with the api direction given. | The test confirms that the ‘year’ variable obtained is equals to ‘ano\_actual’ global variable | The test confirms that the ‘year’ variable obtained is equals to ‘ano\_actual’ global variable | **Pass** |
| 9 | PT9 | GET | PT9\_GET\_validate\_Year\_includes\_  string\_”20” | Validate that the “Year” data includes the string “20” | 1. Add a year parameter, which gets it’s value from a global variable created in the pre-request Script, to the GET instruction  2. There must be a properly configured HTTP request pointing to the endpoint you want to test. | 1. Insert the following pre-request script (it creates a global variable used as parameter):  **var** moment = require('moment');  pm.globals.**set**('ano\_actual', moment().format(("YYYY")) -2);  2. Insert the following script test:  pm.test("Que el 'Year' contenga el string '20'", **function** () {  **var** jsonData = pm.response.json();      pm.expect(jsonData.data[0].Year).to.include("20");  });  3. Execute the GET request with the api direction given. | The test confirms that the ‘Year’ variable obtained is equals to ‘ano\_actual’ global variable | The test confirms that the ‘Year’ variable obtained is equals to ‘ano\_actual’ global variable | **Pass** |
| 10 | PT10 | GET | PT10\_validate\_Pupulation\_is\_above\_  300,000,000 | Validate that the “Population” data is above of 300,000,000 | 1. Add a year parameter, which gets it’s value from a global variable created in the pre-request Script, to the GET instruction  2. There must be a properly configured HTTP request pointing to the endpoint you want to test. | 1. Insert the following pre-request script (it creates a global variable used as parameter):  **var** moment = require('moment');  pm.globals.**set**('ano\_actual', moment().format(("YYYY")) -2);  2. Insert the following script test:  pm.test("Que la población sea mayor de 300,000,000", **function** () {  **var** jsonData = pm.response.json();      pm.expect(jsonData.data[0].Population).to.be.above(300000000);      });  3. Execute the GET request with the api direction given. | The test confirms that the Population is above 300,000,000 | 1. Received a data of 331097593  2. The test confirms that the Population data is above 300,000,000 | **Pass** |