|  |  |
| --- | --- |
| **API** | **Test Cases** |
| Highest Priced Products | 1. Check that response code =200 2. Check that every page have 10 product 3. Check that response includes specified headers 4. Check that price is sorted from high to low |
| Product Name and Description Only | 1. Check that response code =200 2. Check that every page have 10 product 3. Check that response includes name and description only |
| TVs with free shipping and price between $500 and $800 | 1. Check that response code =200 2. Check that every page have 10 product 3. Check that response includes specified headers 4. Check that price is between $500 and $800 and shipping value is 0 |
| Category Names Only | The API IS GIVING INTERNAL SERVER ERROR |
| Categories with TV in the name | 1. Check that response code =200 2. Check that every page have 10 product 3. Check that response includes specified headers 4. Check that every category name contain TV |
| Find stores in Minnesota | 1. Check that response code =200 2. Check that every page have 10 product 3. Check that response includes specified headers |
| Find stores that sell Apple products | 1. Check that response code =200 2. Check that every page have 10 product 3. Check that response includes specified headers |
| Find Stores Within 10 Miles of Beverly Hills | 1. Check that response code =200 2. Check that every page have 10 product 3. Check that response includes specified headers |
| Create Product | 1. Check that response code =200 2. Check that every page have 10 product 3. Check that response includes specified headers 4. Check that request data are equal to request data |
| Create Store | 1. Check that response code =200 2. Check that every page have 10 product 3. Check that response includes specified headers 4. Check that request data are equal to request data |
| Create Service | 1. Check that response code =200 2. Check that every page have 10 product 3. Check that response includes specified headers 4. Check that request data are equal to request data |
| Create Category | 1. Check that response code =200 2. Check that every page have 10 product 3. Check that response includes specified headers 4. Check that request data are equal to request data |

**Test Cases Explanation**

* I used Postman as API Testing tool
* You will see all the test cases per API in the Test tab in Postman
* I will explain each test case code as follows:

1. Check that response code =200

pm.**test**("Status code is 200", **function** () {

    pm.response.to.have.status(200);

    pm.response.to.be.not.empty;

    pm.response.to.be.json;

});

1. Check the Pagination (Example : every page has 10 items)

pm.**test**("data is an Array of 10 products per page", **function** () {

    pm.expect(jsonData.data).to.be.a('array');

    pm.expect(jsonData.data.length).to.be.lessThan(11);

});

1. Check that response includes specified headers (Example : product has **id as a number**)

pm.expect(product).to.have.ownProperty('id');

pm.expect(product.id).to.be.a('number');

1. Check that price is between $500 and $800 and shipping value is 0

pm.expect(product).to.have.ownProperty('price');

pm.expect(product.price).to.be.a('number');

pm.expect(product.price).to.lessThan(801);

pm.expect(product.price).to.greaterThan(499);

pm.expect(product).to.have.ownProperty('shipping');

pm.expect(product.shipping).to.be.a('number');

pm.expect(product.shipping).to.equal(0);

1. Check that every category name contain TV

         pm.expect(catergory).to.have.ownProperty('name');

         pm.expect(catergory.name).to.be.a('string');

       pm.expect(catergory.name).to.contain("TV");

1. For every POST request I user the Pre-request Script to set all the request data as a local variables, so I can verifies that the entered data are equal to the response data

Example of the Pre-request Script to save the request name as a variable:

var reqBody **=** JSON.**parse**(request.data);

pm.variables.**set**("req\_name", reqBody.name);

And then I will compare the request data with the response data as follows:

pm.expect(data).to.have.property('name');

     pm.expect(data.name).to.be.a('string');

var req\_name **=** pm.variables.**get**("req\_name");

pm.expect(data.name).to.equal(req\_name);