

# Postman Code Explanation

## General Instructions:-

1. Lines starting with Keyword “tests” are the condition statements.

For Example: - `tests["Status code is 200"] = responseCode.code === 200;`

It checks whether response status code is 200 OR not.

2. Lines starting with `console.info` are used to print the messages in the Postman Console window.

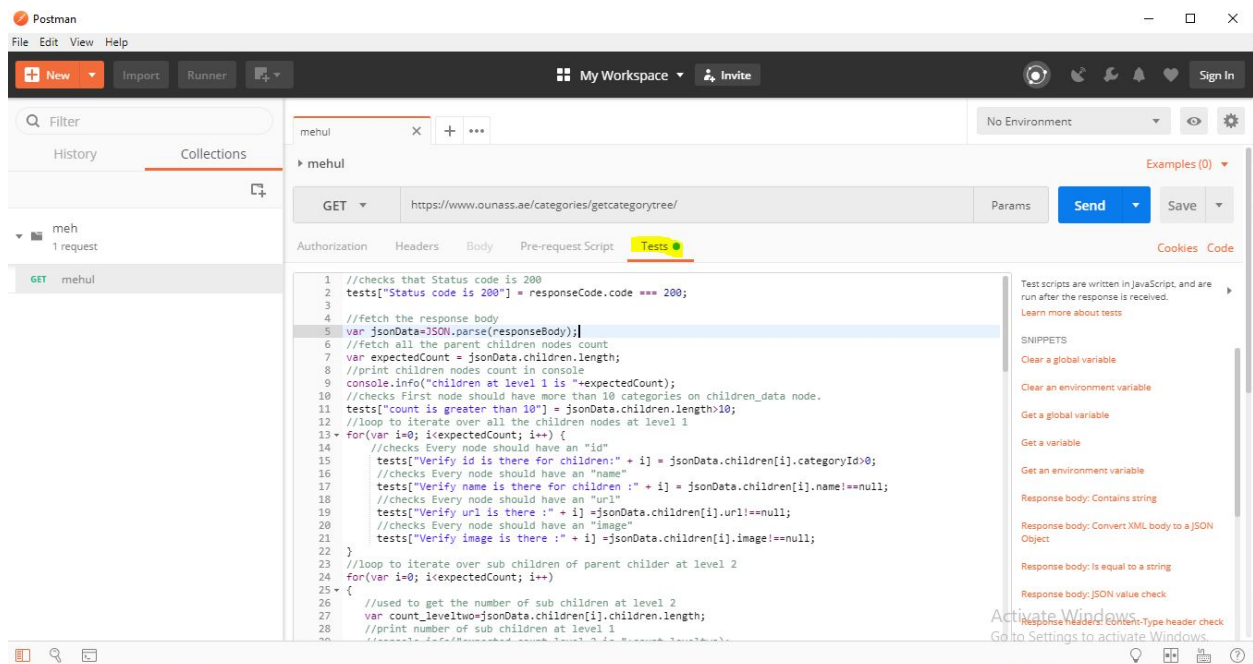
For Example: - `console.info("My name is Mehul");`

It will print “My name is Mehul” in the Postman Console window.

3. To open the Postman Console window navigate to **View -> Show Postman Console**. Here you will see the Console messages.

## Steps to run the script

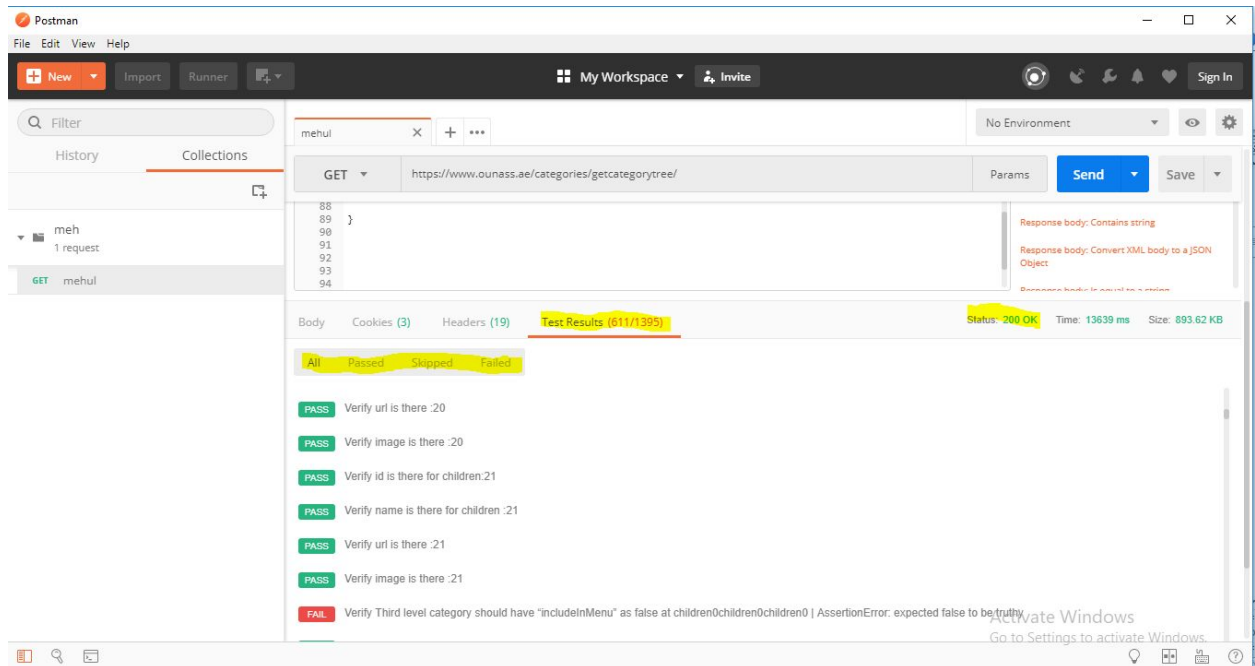
1. As you can see in the diagram, paste the Script in the **Tests(highlighted)** window.



2. Now open the Postman Console. It will look like the below snapshot.



3. Click on Send button to send the Request. Now you will see the Response at the bottom as can be seen in the snapshot below. It indicates status code that is 200 OK, total number of test results Passed (Highlighted).



4. As can be seen in the snapshot below :
  - Pass represents that condition written with Keyword "tests...." is True.
  - Fail represents that condition written with Keyword "tests...." is False.



5. Console outputs can be seen in the Postman Console window as can be seen in the snapshot below.

featureLink is : undefined	05:51:18.797
Verify includeInMenu is falsechildren19children0children2	05:51:18.901
in k	05:51:19.005
	05:51:19.102
length of sortBy0	05:51:19.195
featureLink is : undefined	05:51:19.287
Verify includeInMenu is falsechildren19children0children3	05:51:19.380
in k	05:51:19.471
	05:51:19.563
length of sortBy0	05:51:19.655
featureLink is : undefined	05:51:19.746
Verify includeInMenu is falsechildren19children0children4	05:51:19.837
in k	05:51:19.929
	05:51:20.010
length of sortBy0	05:51:20.103
featureLink is : undefined	05:51:20.201
Verify includeInMenu is falsechildren19children0children5	05:51:20.301
in k	05:51:20.402
expected count level 3 is 0	05:51:20.495
expected count level 2 is 0	05:51:20.586
expected count level 2 is 0	05:51:20.677

## Detailed description of the code

**Condition 1:-** Every response should return 200 OK.

**Result:** - The below lines check that Status code is 200

**//checks that Status code is 200**

```
tests["Status code is 200"] = responseCode.code === 200;
```

**Condition 2:-** First node should have more than 10 categories on children\_data node.

**Result:** - The below lines checks “First node should have more than 10 categories on children\_data node.”

```
//fetch the response body
```

```
var jsonData=JSON.parse(responseBody);
```

```
//fetch all the parent children nodes count
```

```
var expectedCount = jsonData.children.length;
```

```
//print children nodes count in console
```

```
console.info("children at level 1 is "+expectedCount);
```

```
//checks First node should have more than 10 categories on children_data node.
```

```
tests["count is greater than 10"] = jsonData.children.length>10;
```

**Condition 3:-**

- Every node should have an "id"
- Every node should have a "name"
- Every node should have a "url"
- Every node should have an "image" except 3rd level category

**Result:** - The below mentioned lines checks the above conditions.

```
//loop to iterate over all the children nodes at level 1
```

```
for(var i=0; i<expectedCount; i++)
```

```
{
```

```
//checks Every node should have an "id"
```

```
tests["Verify id is there for children:" + i] = jsonData.children[i].categoryId>0;
```

```
//checks Every node should have an "name"
```

```
tests["Verify name is there for children :" + i] = jsonData.children[i].name!==null;
```

```

//checks Every node should have an "url"

tests["Verify url is there :" + i] =jsonData.children[i].url!==null;

//checks Every node should have an "image"

tests["Verify image is there :" + i] =jsonData.children[i].image!==null;

}

```

#### Condition 4:-

- Third level category should have “includeInMenu” as false
- Last level of categories should have “availableSortBy” minimum of 2 sub nodes;
  - shoeSizeCode
  - designer

**Result:** - The below mentioned lines checks the above conditions.

```

//loop to iterate over sub children of parent children at level 2

for(var i=0; i<expectedCount; i++)
{
    //used to get the number of sub children at level 2

    var count_leveltwo=jsonData.children[i].children.length;

    //print number of sub children at level 1

    //console.info("expected count level 2 is "+count_leveltwo);

    //loop to iterate over sub children at level 3.

    for(var j=0;j<count_leveltwo;j++)
    {
        //used to get the number of sub children at level 3

        var count_levelthree=jsonData.children[i].children[j].children.length;

        //loop to iterate over sub children at level 3.

        for(var k=0;k<count_levelthree;k++)
        {
            //used to get the response body of children at level 3.

```

```

var jsonObject=jsonData.children[i].children[j].children[k];

// var image=jsonObject.image;

//used to get includeInMenu value at level 3.

var includeInMenu=jsonObject.includeInMenu;

//used to get availableSortBy value elements in it at level 3.

var sortBySize=jsonObject.availableSortBy.length;

//used to print number of availableSortBy elements in it at level 3.

console.info("length of sortBy"+jsonObject.availableSortBy.length);

//used to get featureLink value at level 3.

var featureLink=jsonObject.featureLink;

//console.info("featureLink is : "+jsonObject.featureLink);

//verify Third level category should have "includeInMenu" as false

tests["Verify Third level category should have "includeInMenu" as false at "+"children
"+"i+"children "+"j+"children "+"k"] =includeInMenu===false;

if(includeInMenu===false)

{

    //print whether Third level category should have "includeInMenu" as false

    console.info("Verify includeinMenu is false at "+"children "+"i+"children "+"j+"children "+"k);

}

// check Last level of categories should have "availableSortBy" minimum of 2 sub nodes

tests["Verify sortBySize has min 2 sub nodes : at "+"children "+"i+"children "+"j+"children "+"k]
=sortBySize>=2;


// check Last level of categories should have "availableSortBy" minimum of 2 sub nodes

if(sortBySize>=2)

{

    //print whether Last level of categories should have "availableSortBy" minimum of 2 sub nodes

```

```

    console.info("sortBySize has minimum of 2 subnodes for "+"children "+i+"children "+j+"children
"+k);

    }

    //to convert object to string

    var flink="" +featureLink;

    //checks whether A node with featureLink with “/edits/*” should be in 3rd level

    tests["Verify featureLink is there : "+"children "+i+"children "+j+"children "+k]
    =flink.includes("/edits/");

    //checks whether A node with featureLink with “/edits/*” should be in 3rd level

    if(flink.includes("/edits/"))

    {

        //print whether A node with featureLink with “/edits/*” should be in 3rd level

        console.info("featureLink has /edits/* for "+"children "+i+"children "+j+"children "+k);

    }

    }

}

}

```