

website of fake API

<https://fakestoreapi.com/docs?ref=insights.itexto.com.br#tag/Users>



USERS




Type of API : GET


Test script for GET all users


```
pm.test("Status code is 200", function () {  
  
    pm.response.to.have.status(200);  
  
});  
  
pm.test("Response contains users", function () {  
  
    var jsonData = pm.response.json();  
  
    pm.expect(jsonData.length).to.be.above(0);  
  
});  
  
console.log("First user:", pm.response.json()[0]);
```

result of test script:

Test Results  

200 OK • 579 ms • 1.56 KB •  |  Save Response 

Filter Results 



PASSED Status code is 200

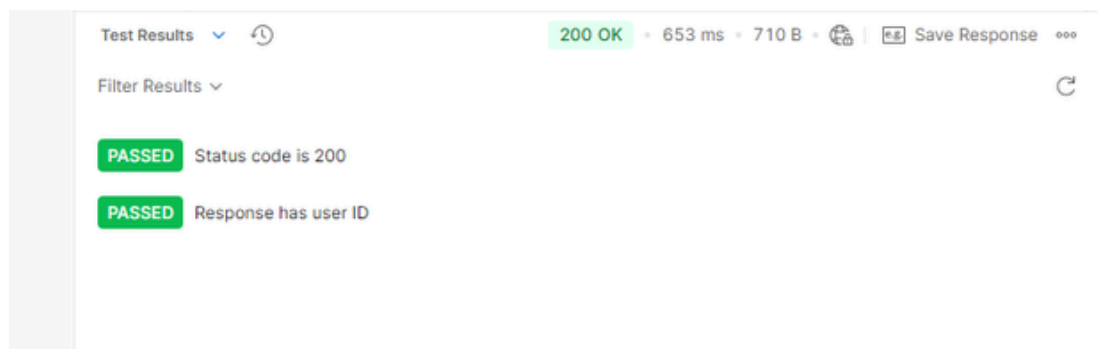
PASSED Response contains users

Type of API : POST

test script for add a new user

```
pm.test("Status code is 200", function () {  
  
    pm.response.to.have.status(200);  
  
});  
  
pm.test("Response has user ID", function () {  
  
    var jsonData = pm.response.json();  
  
    pm.expect(jsonData).to.have.property("id");  
  
    console.log("New user ID:", jsonData.id);  
  
});
```

result of test script

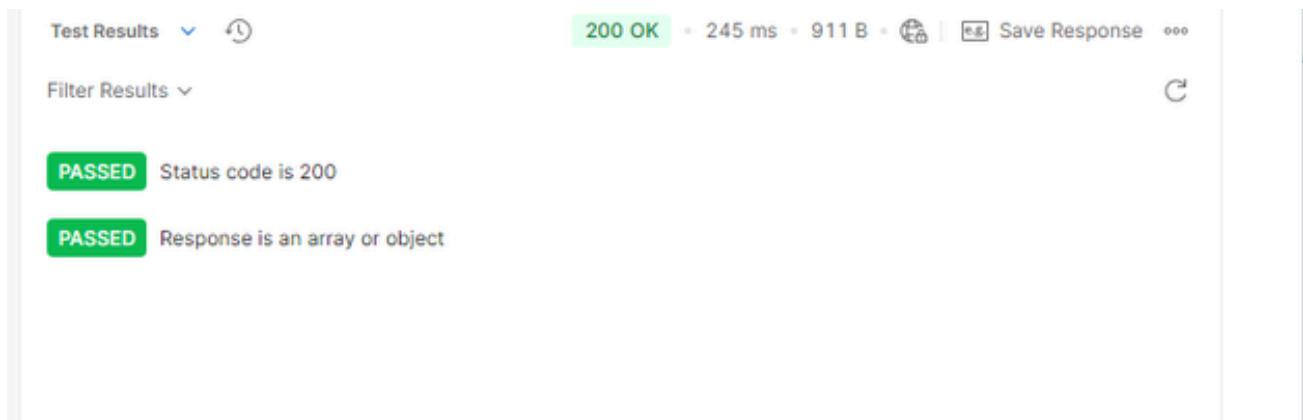


type of API : GET

Test script for get single user

```
pm.test("Status code is 200", function () {  
  
    pm.response.to.have.status(200);  
  
});  
  
pm.test("Response is an array or object", function () {  
  
    var jsonData = pm.response.json();  
  
    pm.expect(typeof jsonData === "object").to.be.true;  
  
});  
  
console.log("Response body:", pm.response.json());
```

result of test script

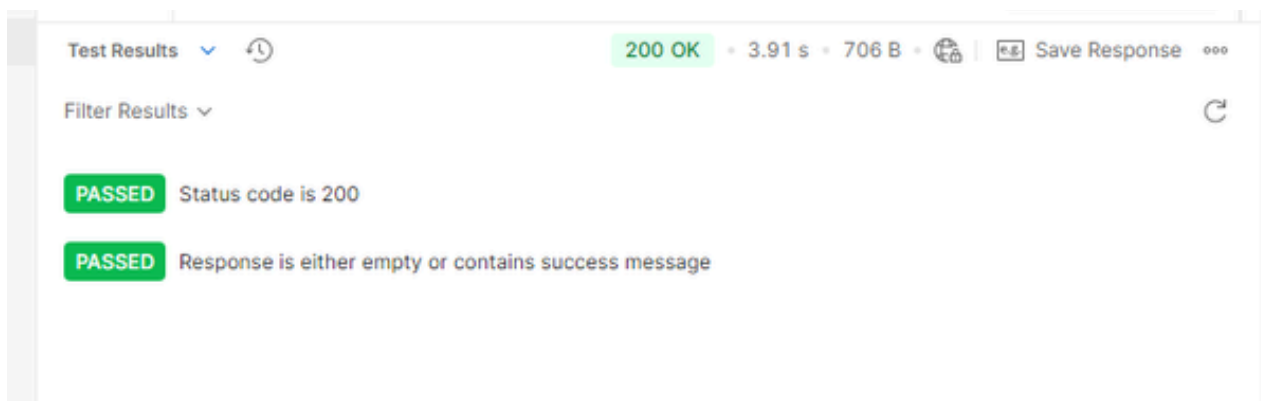


type of API : PUT

test script for update a user

```
pm.test("Status code is 200", function () {  
  
    pm.response.to.have.status(200);  
  
});  
  
pm.test("Response is either empty or contains success message", function () {  
  
    var jsonData = pm.response.json();  
  
    console.log("Response:", jsonData);  
  
    pm.expect(jsonData).to.be.an('object');  
  
});
```

result of test script



type of API : DELETE

test script for delete a user

```
pm.test("Status code is 200 or 400", function () {

    pm.expect(pm.response.code).to.be.oneOf([200, 400]);

});

pm.test("Response message for 200 status", function () {

    var jsonData = pm.response.json();

    if (pm.response.code === 200) {

        console.log("User deleted successfully:", jsonData);

        pm.expect(jsonData).to.have.property('id').that.is.a('number');

        pm.expect(jsonData).to.have.property('address').that.is.an('object');

    }

});

pm.test("Response message for 400 status", function () {

    var jsonData = pm.response.json();

    if (pm.response.code === 400) {


        console.log("Bad Request Error:", jsonData);

        pm.expect(jsonData).to.have.property('message');

    }

});
```

result of test script

Test Results  

200 OK • 240 ms • 917 B •  |  Save Response ...

Filter Results 



PASSED Status code is 200 or 400

PASSED Response message for 200 status

PASSED Response message for 400 status