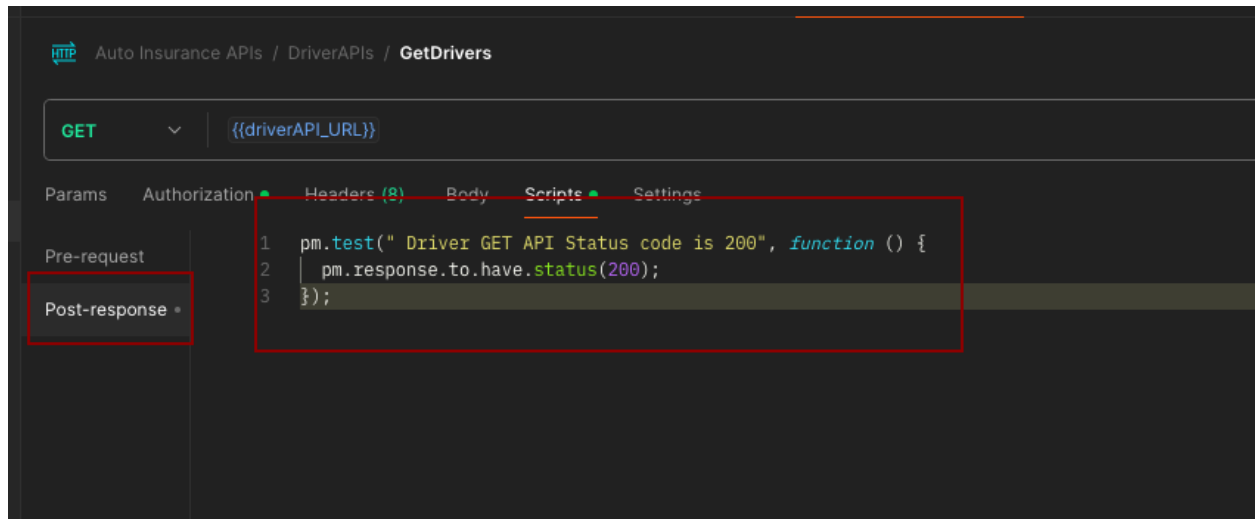


Automation Test Scripts

Step #1: Create a new Driver API request such as Driver GET API or Driver Post API

Step #2: Inside the Script - Post-Request tab enter the script for your test cases such as "Status Code should be 200".



Step #3: hit send and make sure test results are expected.

Step #4: Create similar test cases for different scenarios and automate them. Use the below script code.

Driver API Automation Postman functions

Test Scenario: To test if status code is 200

```
pm.test(" Driver GET API Status code is 200", function () {  
  pm.response.to.have.status(200) ;  
});
```

Test Scenario: To test if the status code is one of the list.

```
pm.test("Status code is 200, 201, 202", function () {  
  pm.expect(pm.response.code).to.be.oneOf([200,201, 202]);  
});
```

Test Scenario: //To Test if the firstname for id '63ebdf59aa86075000059eb8' is "Float!"

```
pm.test("First name is Float!", () => {  
  const responseJson = pm.response.json();  
  pm.expect(responseJson.firstname).to.eql("Float!");  
});
```

Test Scenario: //To Test if the ssn for id '63ebdf59aa86075000059eb8' is 851-45-6733"

```
pm.test("ssn is 851-45-6733", () => {  
  const responseJson = pm.response.json();  
  pm.expect(responseJson.ssn).to.eql("851-45-6733");  
});
```

Test Scenario: //To Test if the firstname for id '63c19ea2dab2072b00012a66' is Krystina"

```
pm.test("First name is Krystina", () => {  
  const responseJson = pm.response.json();  
  pm.expect(responseJson.firstname).to.eql("Krystina");  
});
```

Vehicle Get API Automation

Test Scenario: //Status code is 200

```
pm.test("Vehicle - Status code is 200", function () {  
  pm.response.to.have.status(200);  
});
```

Test Scenario: //To Test if Vehcile Year for id '63c2a299dab2072b000139d7' is 2020-01-01

```
pm.test("Vehcile year is 2020", () => {  
  const responseJson = pm.response.json();  
  pm.expect(responseJson.year).to.eql("2020-01-01T00:00:00.000Z");  
});
```

Test Scenario: //To Test if Vehicle Make is Lincoln for id '63c2a299dab2072b000139d7'

```
pm.test("Vehicle Make is Lincoln", () => {  
  const responseJson = pm.response.json();  
  pm.expect(responseJson.Make).to.eql("Lincoln");  
});
```

Test Scenario: //To Test if Vehicle Trim is LTZ for id '63c2a299dab2072b000139d7'

```
pm.test("Vehicle Trim is Trim", () => {  
  const responseJson = pm.response.json();  
  pm.expect(responseJson.Trim).to.eql("LTZ");  
});
```

Test Scenario: //To Test if Mileage is 23233 for id '63c2a299dab2072b000139d7'

```
pm.test("Vehicle Mileage is Mileage", () => {  
  const responseJson = pm.response.json();  
  pm.expect(responseJson.Mileage).to.eql(23233);  
});
```

Test Scenario: //To Test if VIN is 69527781668 for id '63c2a299dab2072b000139d7'

```
pm.test("Vehicle Mileage is VIN", () => {  
  const responseJson = pm.response.json();  
  pm.expect(responseJson.vin).to.eql(69527781668);  
});
```

Other Postman Automation functions

Test Scenario: //Status code validation

```
pm.test("Status code is 200", function () {  
  pm.response.to.have.status(200);  
});
```

Test Scenario: //First Name Validation

```
pm.test("First name validation", () => {  
  //parse the response JSON and test three properties  
  const responseJson = pm.response.json();  
  pm.expect(responseJson.firstname).to.eql(pm.environment.get("driver_firstname"));
```

```
});
```

Test Scenario: //Last Name validation

```
pm.test("Last name validation", () => {  
  //parse the response JSON and test three properties  
  const responseJson = pm.response.json();  
  pm.expect(responseJson.lastname).to.eql(pm.environment.get("driver_lastname"));  
});
```

Test Scenario: //Test if Status has Created in it

```
pm.test("Status code name has string", () => {  
  pm.response.to.have.status("Created");  
});
```

Test Scenario: //Test if entire body has word 'Float' in it

```
pm.test("Body is string", function () {  
  pm.response.to.have.body("Float");  
});
```

Test Scenario: /Response time

```
pm.test("Response time is less than 200ms", () => {  
  pm.expect(pm.response.responseTime).to.be.below(200);  
});
```

Test Scenario: //Check if value is from the list

```
pm.test("Value is in valid list", () => {  
  pm.expect(pm.response.json().title)  
    .to.be.oneOf(["Mr", "Mr", "MS"]);  
});
```

Test Scenario: //Log a value

```
console.log(pm.collectionVariables.get("firstname"));  
console.log(pm.response.json().lastname);
```

Test Scenario: //Custom validation

```
if (pm.response.json().id) {  
  console.log("id was found!");  
  // do something  
} else {  
  console.log("no id ...");  
}
```

```
//do something else
```

```
}
```