**Documentation and test cases**

Below mentioned are the test scripts for GET request to retrieve the pets which are in sold status.

* **pm** is the object which comes from Postman’s PM API (known as the pm.\* API)
* **pm.test()** function is used to write test specifications refer first Test case)

**It accepts 2 parameters**

1. name of the test as a string
2. a function to return a Boolean value

* **pm.expect()** is assertion function used to validate the result from an API. which comes from JavaScript test library called ChaiJS BDD (refer Second Test case)

**//First Test Case**

**//Validate the status code**

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

pm.response.to.have.status(200); **// returns true if the status code is 200**

});

**//Second Test Case**

Below code validates the API response time, pm.expect() is assertion

**//Validate response time is below 300ms**

pm.test("Response time is less than 300ms", function () {

pm.expect(pm.response.responseTime).to.be.below(300); **// returns true if the response time is below 300**

});

**//Third Test Case**

**//Validate API schema**

pm.test('Validate the Schema', function () {

pm.expect(tv4.validate(pm.response.json(), JsonSchema)).to.be.true; **// returns true if schema is correct**

});

**Test Cases:**

**Prerequisites :**

**Swagger:** <https://petstore.swagger.io/#/>

**API description** : Finds Pets by status

**Request Type** : **GET**

**URI** : **pets/findByStatus**

**Request URL** : https://petstore.swagger.io/v2/pet/findByStatus?status=?

\*Status is a require field:   
**Available status are :**

1. available
2. pending
3. sold

ex : https://petstore.swagger.io/v2/pet/findByStatus?status=sold

**Headers :**

**Content-Type** : application/json

**Test Case 1 :**

1. Execute GET request using the request URL with status as ‘**available’** as a parameter
2. Validate the status/ response code as 200.
3. Validate the API response for correct results
4. Compare the API response with Data base result set
5. Repeat from step 1 to 4 for other statuses [pending and sold] and validate the response code and status also with data base results

**Test Case 2:**

1. Execute GET request with empty parameter
2. Validate the response code as 400
3. Validate the response with error message :“Status value must not be ‘null’ : Possible parameters are :“available,pending,sold”

ex : https://petstore.swagger.io/v2/pet/findByStatus?status=null

**Test Case 3:**

1. Execute GET request with wrong parameter
2. Validate the response code as 400
3. Validate the response with error message :“No resources found : Possible parameters are :“available,pending,sold”

ex : https://petstore.swagger.io/v2/pet/findByStatus?status=status