**COURSE-END PROJECT: NON-FUNCTIONAL TESTING USING POSTMAN, REST ASSURED, AND JMETER.**

**DESCRIPTION:**

To automate functionalities using <https://petstore.swagger.io/> REST API services

**SCENARIO:**

 You are working as a Test Engineer in XYZ Corp. Your company has decided to automate a few functionalities for one of the Pet Store companies.

You have been asked to design an end-to-end functionality to automate three REST API services using Postman and Rest Assured.

**TOOLS REQUIRED:**

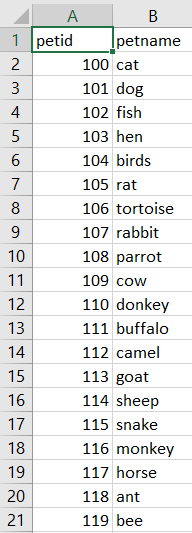
* Postman
* Java 1.8+
* Maven
* Rest Assured Maven dependency version 4.5.1
* TestNG Maven dependency version 7.1.0
* Hamcrest Maven dependency
* Newman for Postman
* JMeter

**GITHUB URL:** [**https://github.com/Vijay-Vel/phase3-EndProject**](https://github.com/Vijay-Vel/phase3-EndProject)

**POSTMAN ASSIGNMENT\_001:**

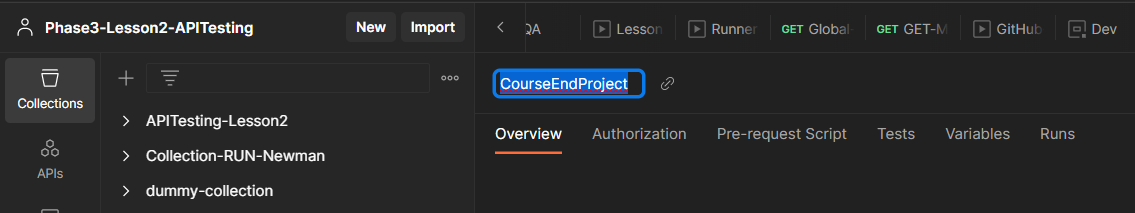
**projectdata.csv:**

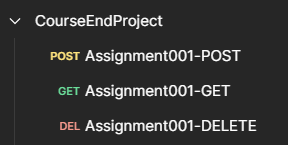
● Created a CSV file with two columns, petID and petName, having 20 rows’ details



● Open Postman Application

● Create a new collection called **CourseEndProject** and arrange three end-to-end services POST





● In the collection, add a new POST service request with following details: ○ URL: <https://petstore.swagger.io/v2/pet>

○ Service Type: POST

○ JSON Body:

{

"id": “{{petid}}”,

"category": {

"id": 0,

"name": "string"

},

"name": "{{petname}}",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

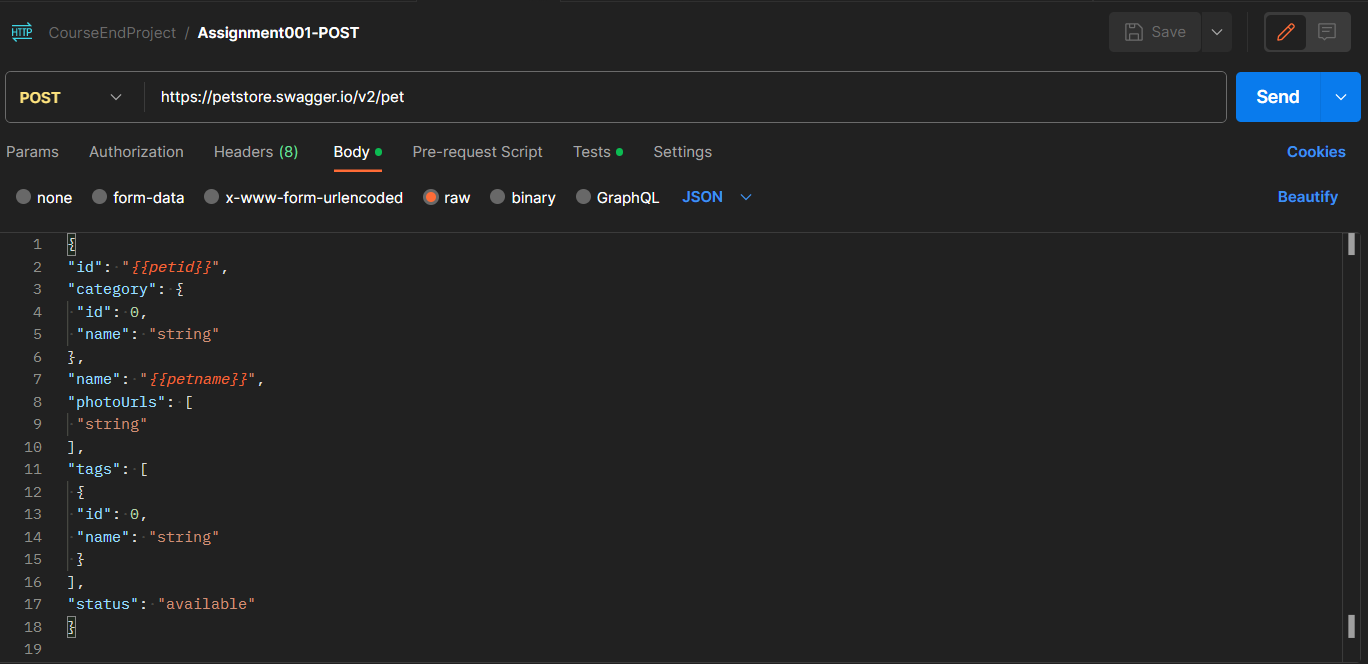
"name": "string"

}

],

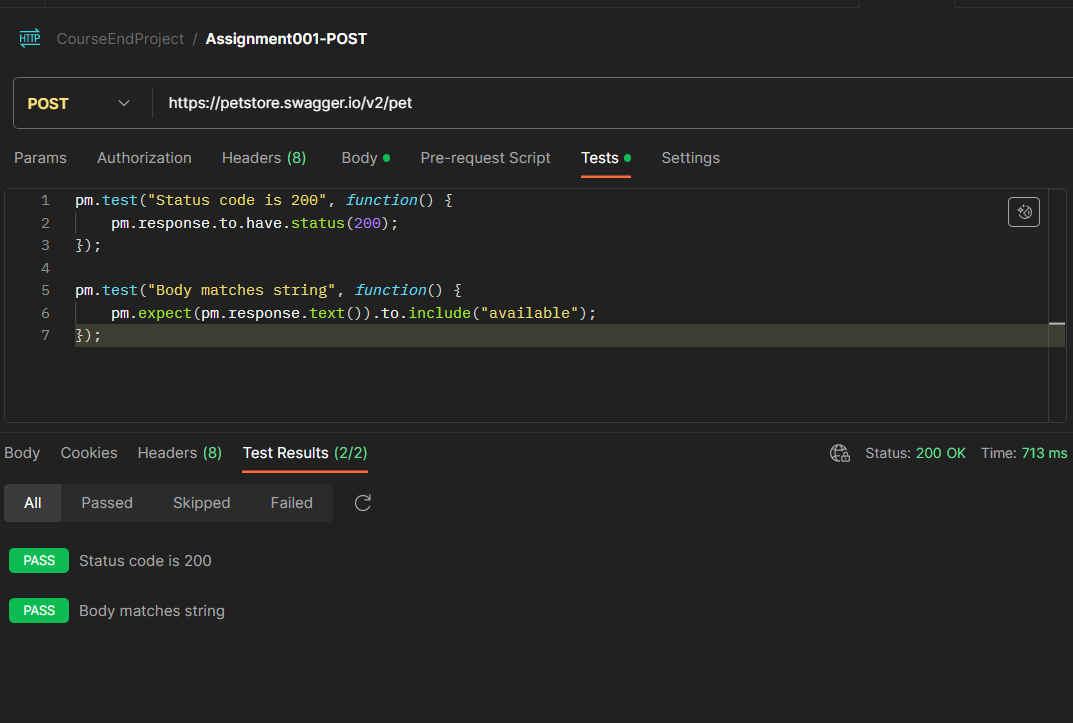
"status": "available"

}

****

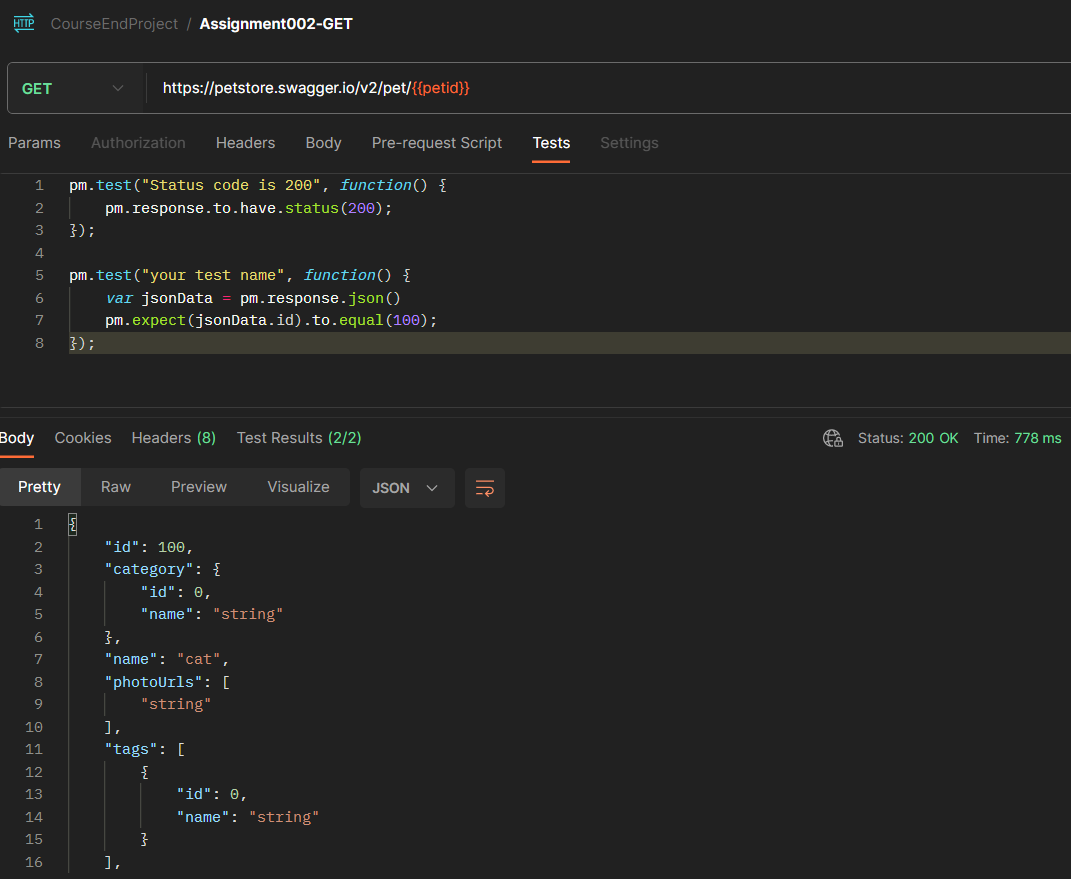
● In the Postman Tests section, validate for this POST call. Response status code should be 200.

● In the Postman Tests section validate for this POST call. Response body should contain text as **‘available’**.

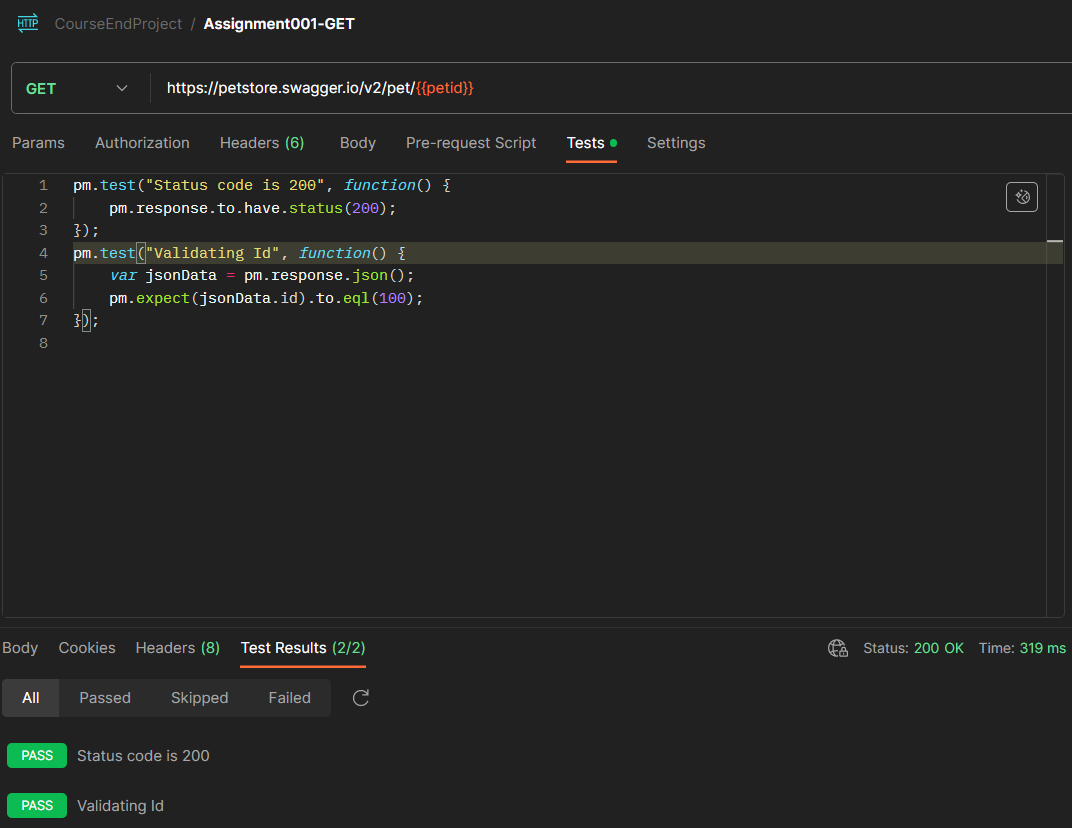


**Validate petID:**

* 1. • In the collection, add a new GET service request with the following details: o URL: <https://petstore.swagger.io/v2/pet/2003>



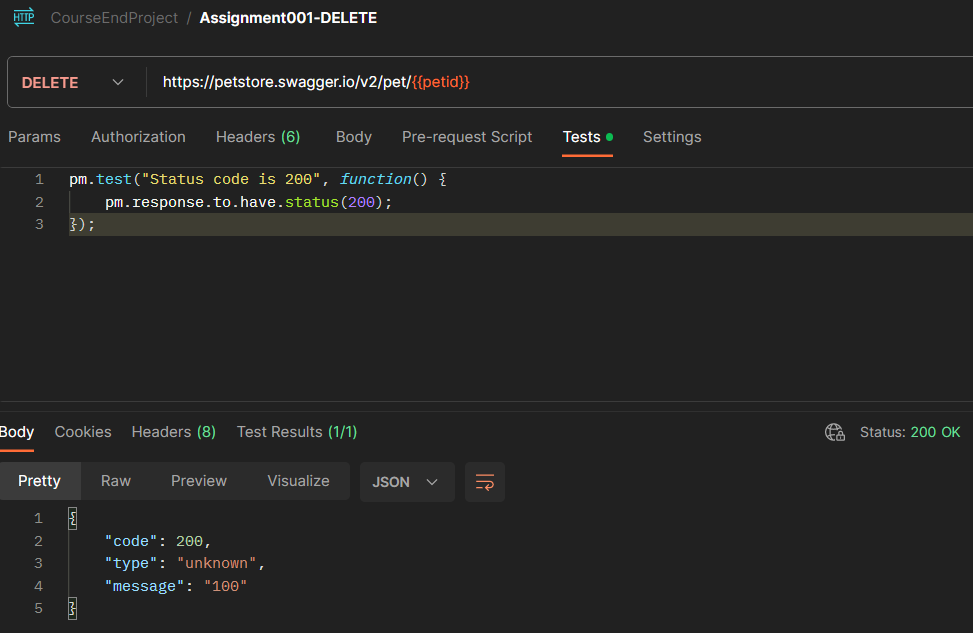
* 1. • In the service URL pet **id**, which is highlighted in YELLOW color, should be parameterized in Postman and runtime data should be driven through the CSV file.
  3. • In the Postman Tests section validate for this GET call. Response status code should be 200.



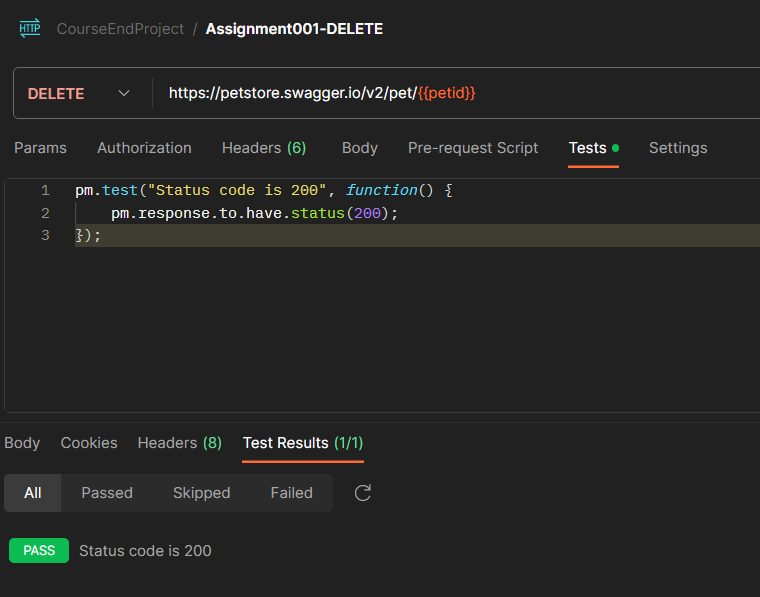
**Delete petID:**

● In the collection, add a new DELETE service request with the following details: ○ URL: <https://petstore.swagger.io/v2/pet/2003>

● In the service URL pet **id**, which is highlighted in YELLOW color should be parameterized in Postman and runtime data should be driven through the CSV file.

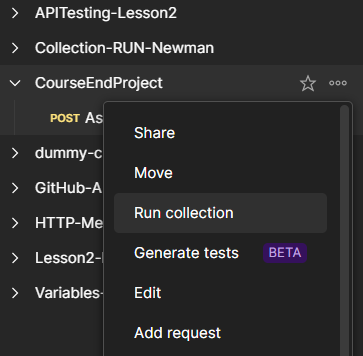


● In the Postman Tests section validate for this GET call. Response status code should be 200.

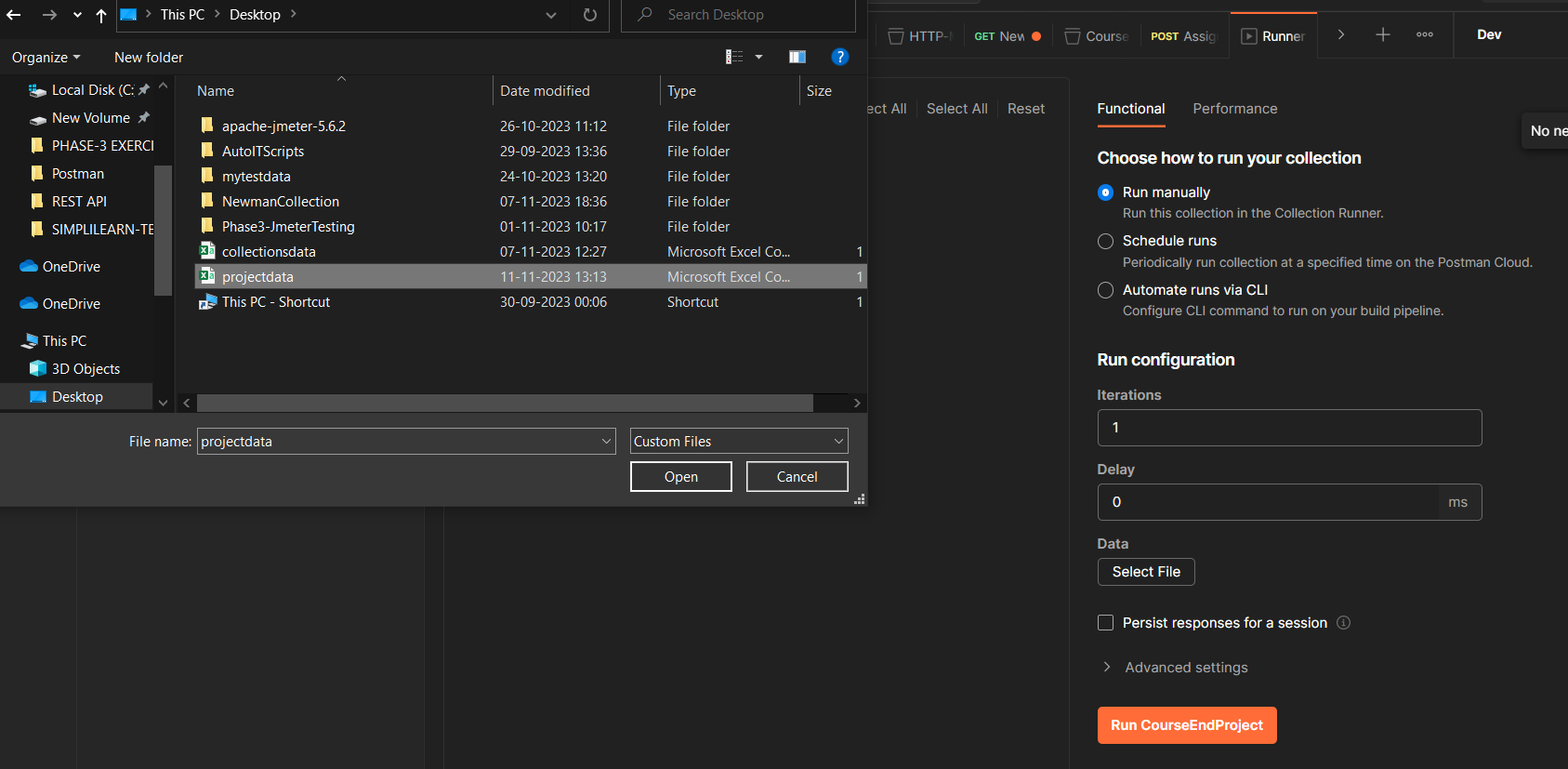


**Steps to Run:**

1. Run collection

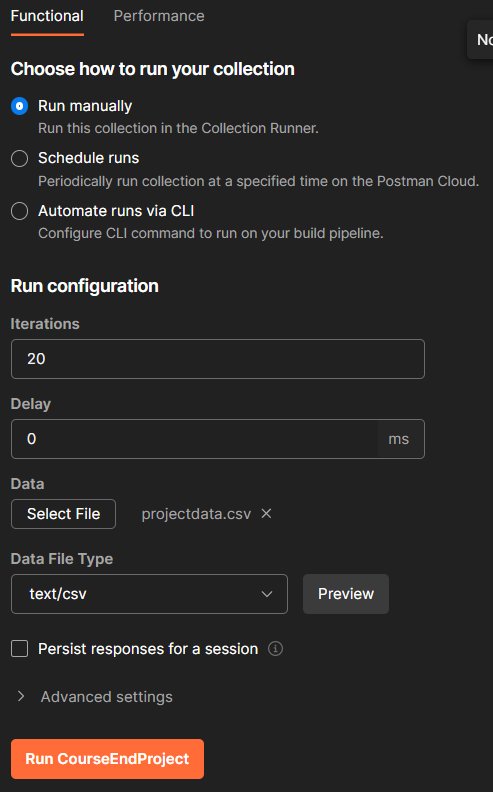


1. Select the CSV file to run these End-to-End scenarios 20 times

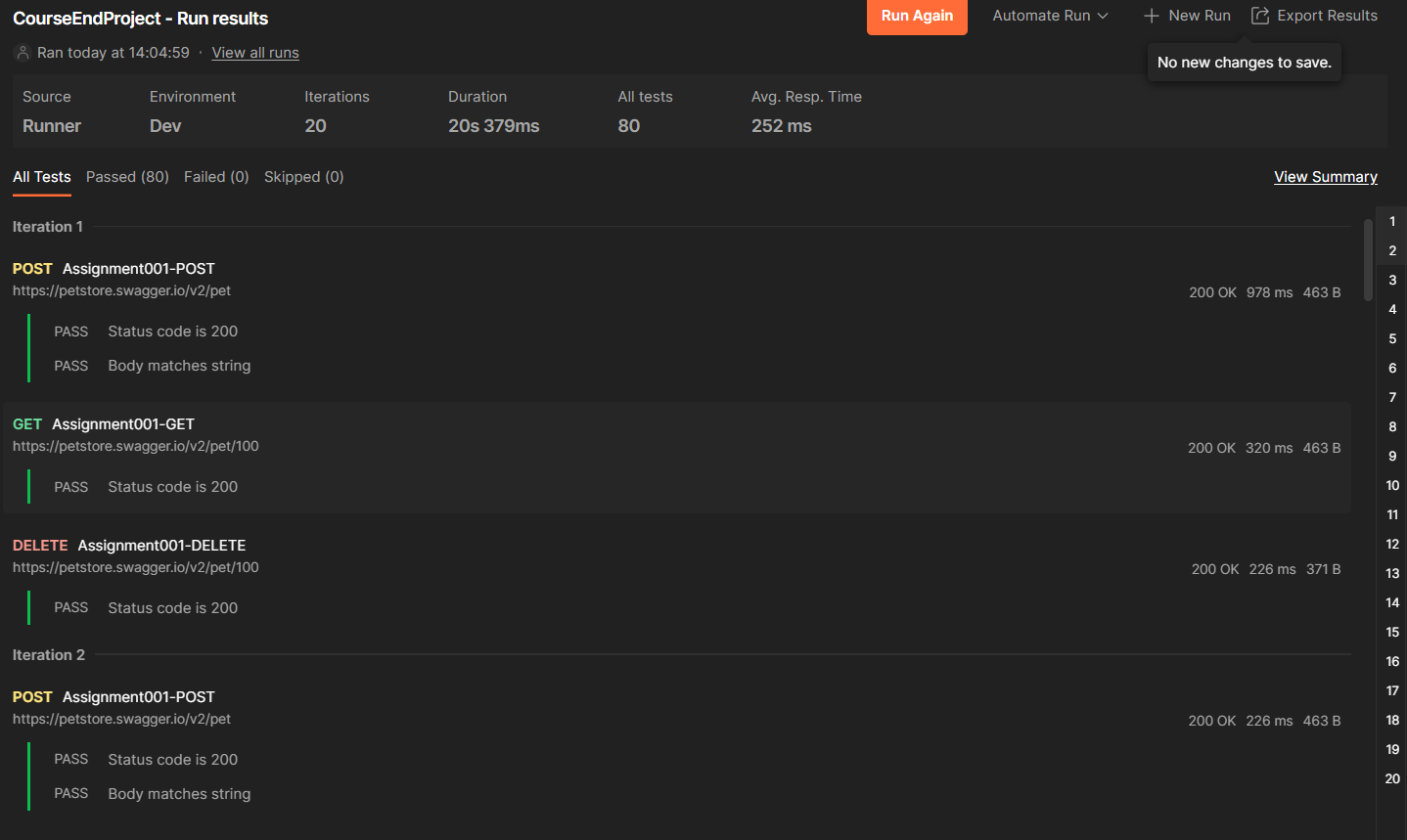


3. Check the checkbox for Save Response

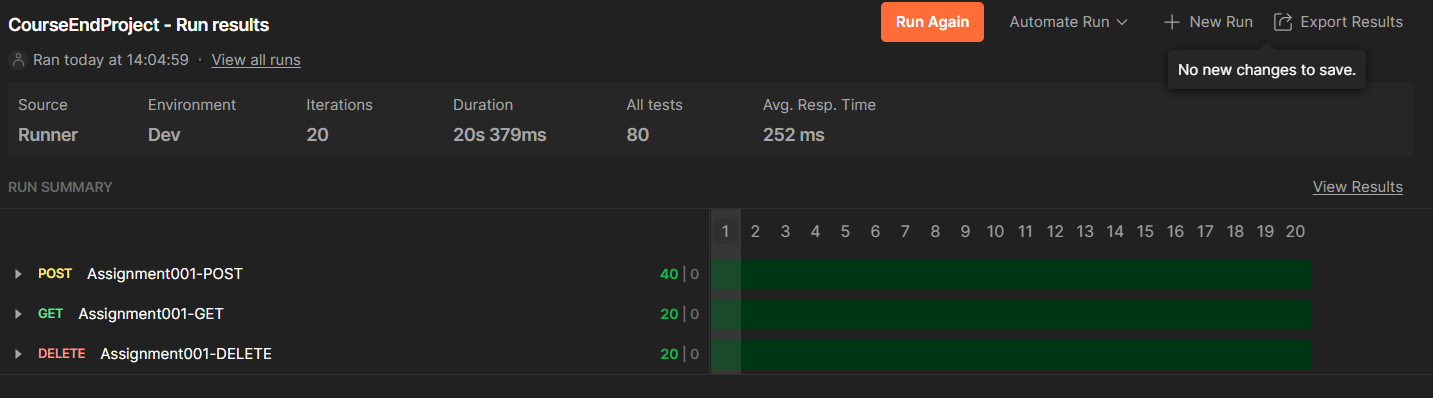
4. Open Postman Console and Run the collection



5. Validate that all parameterized data [PetID/PetName] should be filled with CSV runtime data

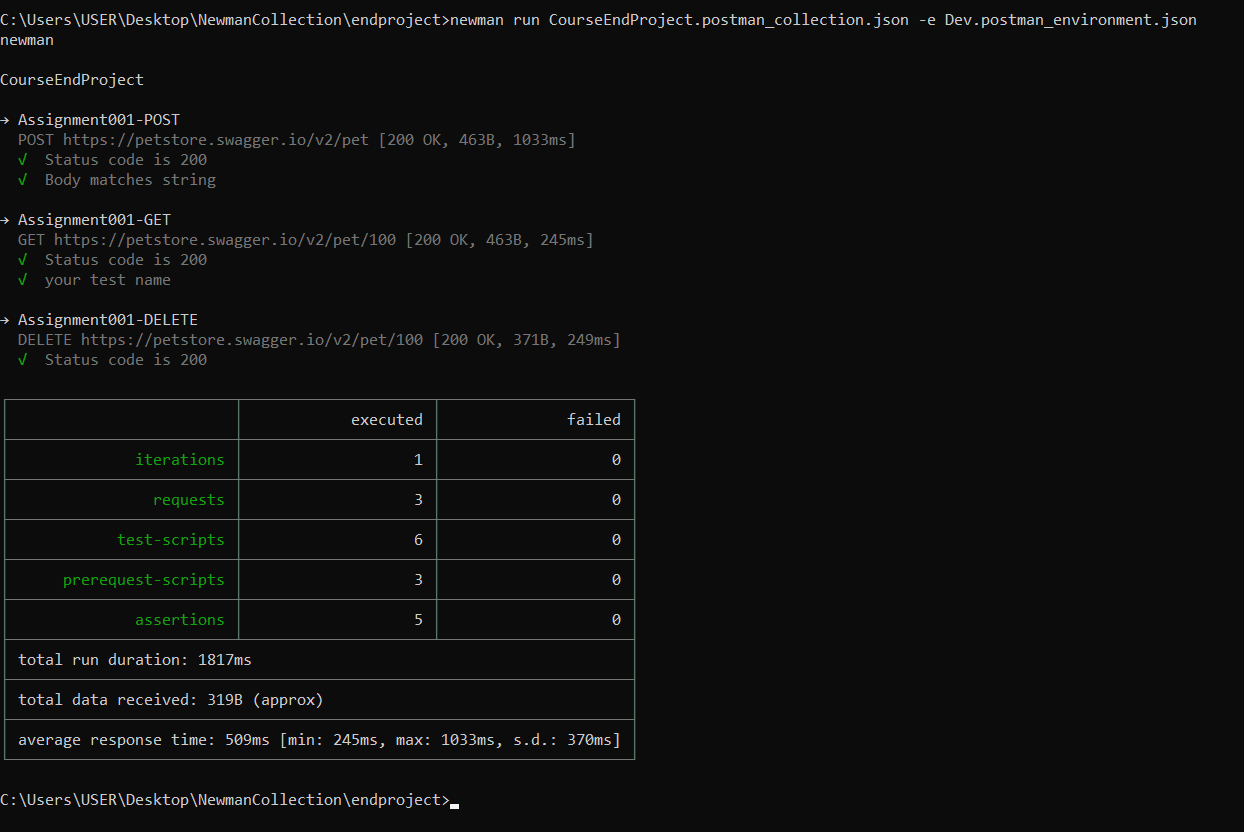


6. All status codes should pass



7. Export the collection as a JSON file

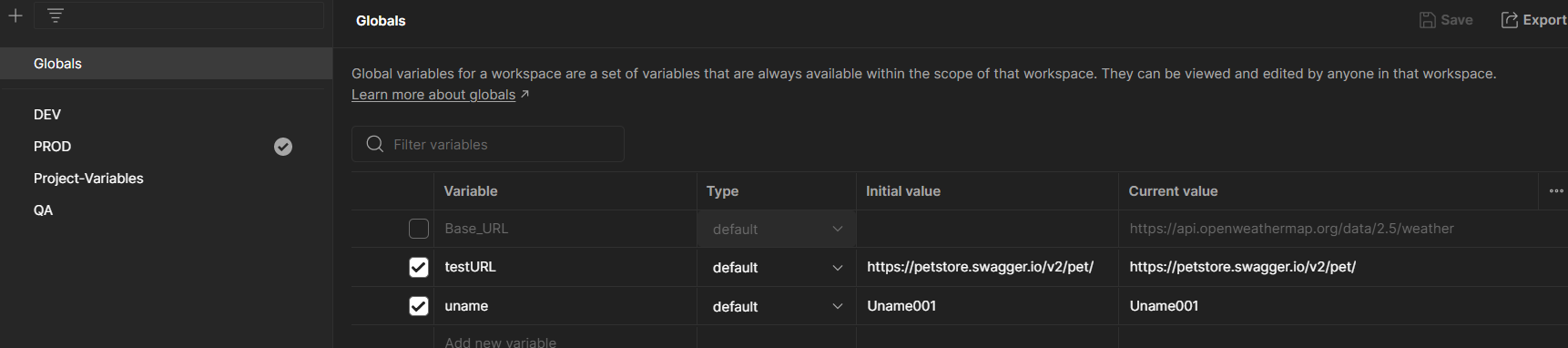
8. Run this JSON collection using the Postman Newman command from cmd/terminal.



**POSTMAN ASSIGNMENT\_002:**

● Hit a **PUT** call having service URL = <https://petstore.swagger.io/v2/pet>

● Create a global variable of this URL where **testURL** is the variable name and its value is https://petstore.swagger.io/v2/pet. Use this variable while hitting the URL in Postman.



● PUT Call request JSON body:

{

"id": 9223372016900013000, "category": {

"id": 20021,

"name": "string" },

"name": "doggie", "photoUrls": [

"string"

], "tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available\_QA"

}

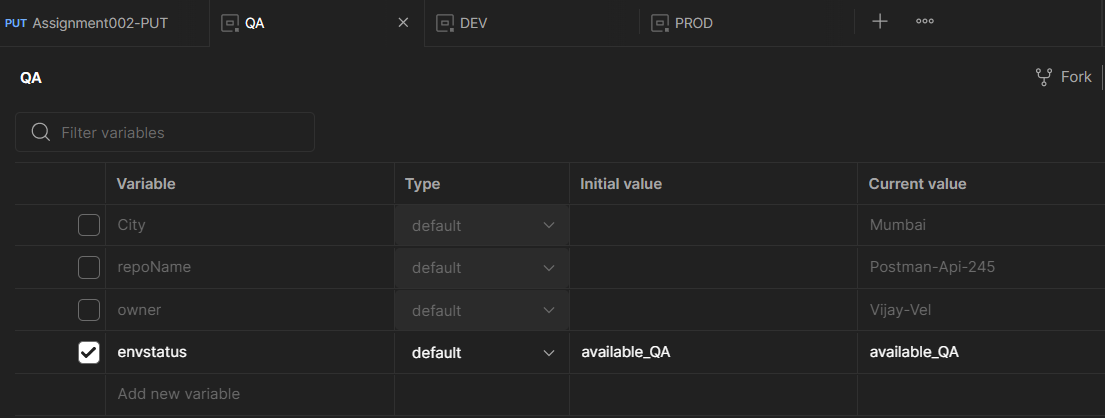
● Create 3 test Environments as DEV, QA, PROD. The PUT call JSON body **job** field should be parameterized and its value should change as per environment:

○ When Environment is DEV then "status": "available\_DEV"

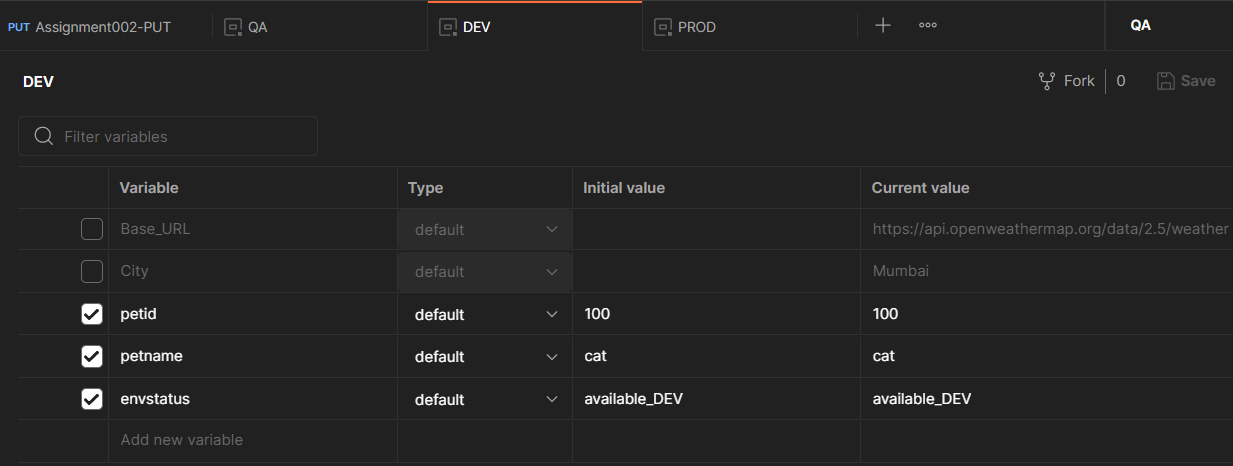
○ When Environment is QA then "status": "available\_QA"

○ When Environment is PROD then "status": "available\_PROD"

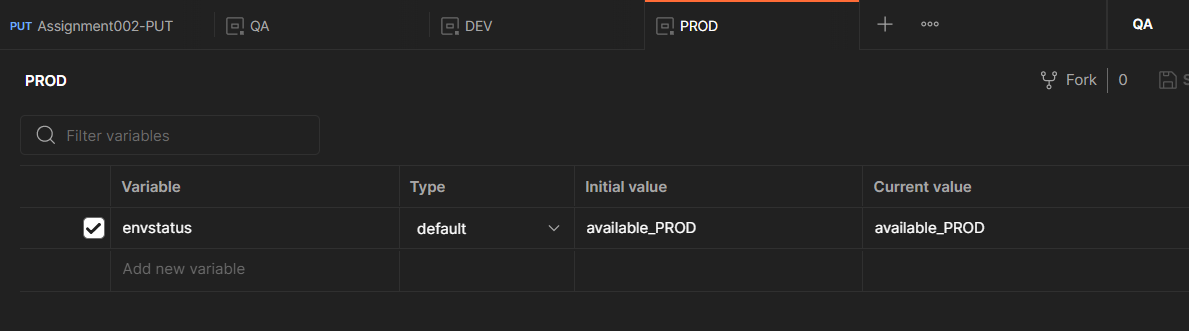
**QA ENVIRONMENT**



**DEV ENVIRONMENT**

****

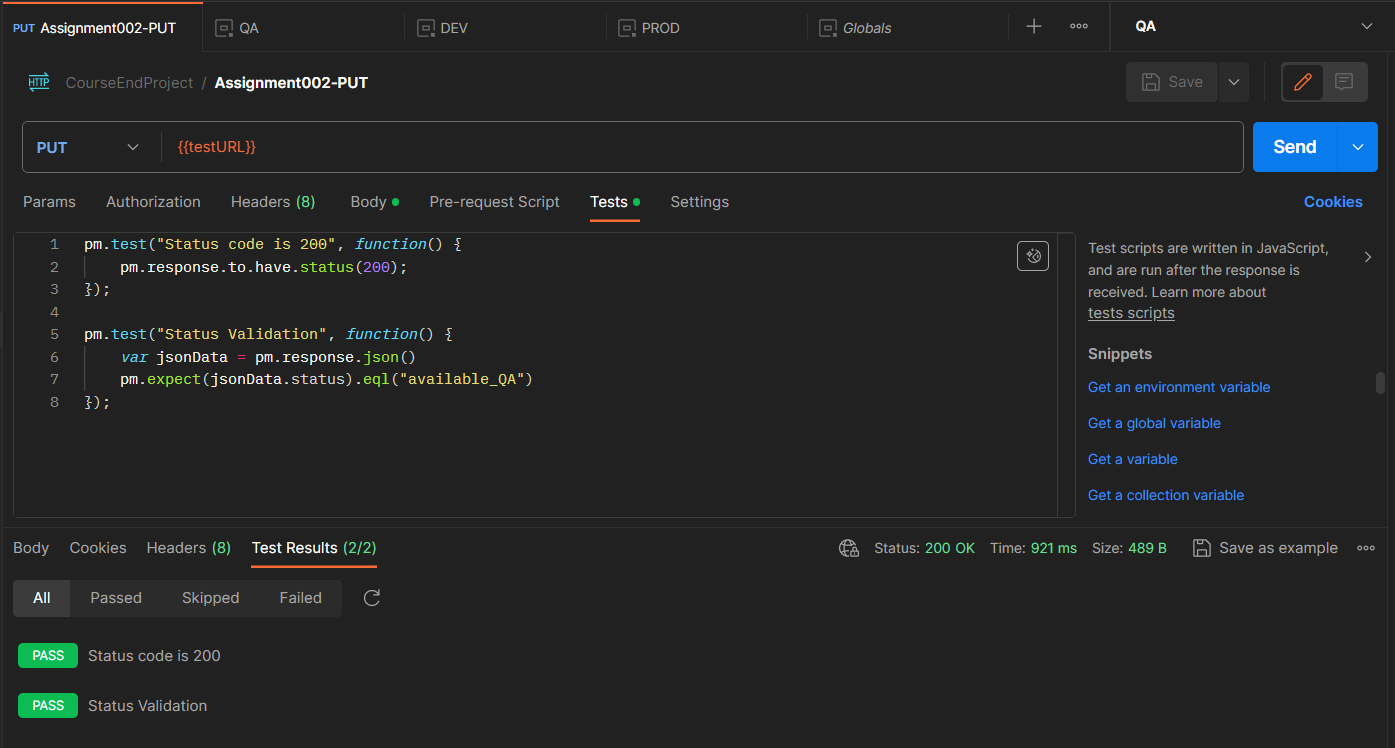
**PROD ENVIRONMENT**

****

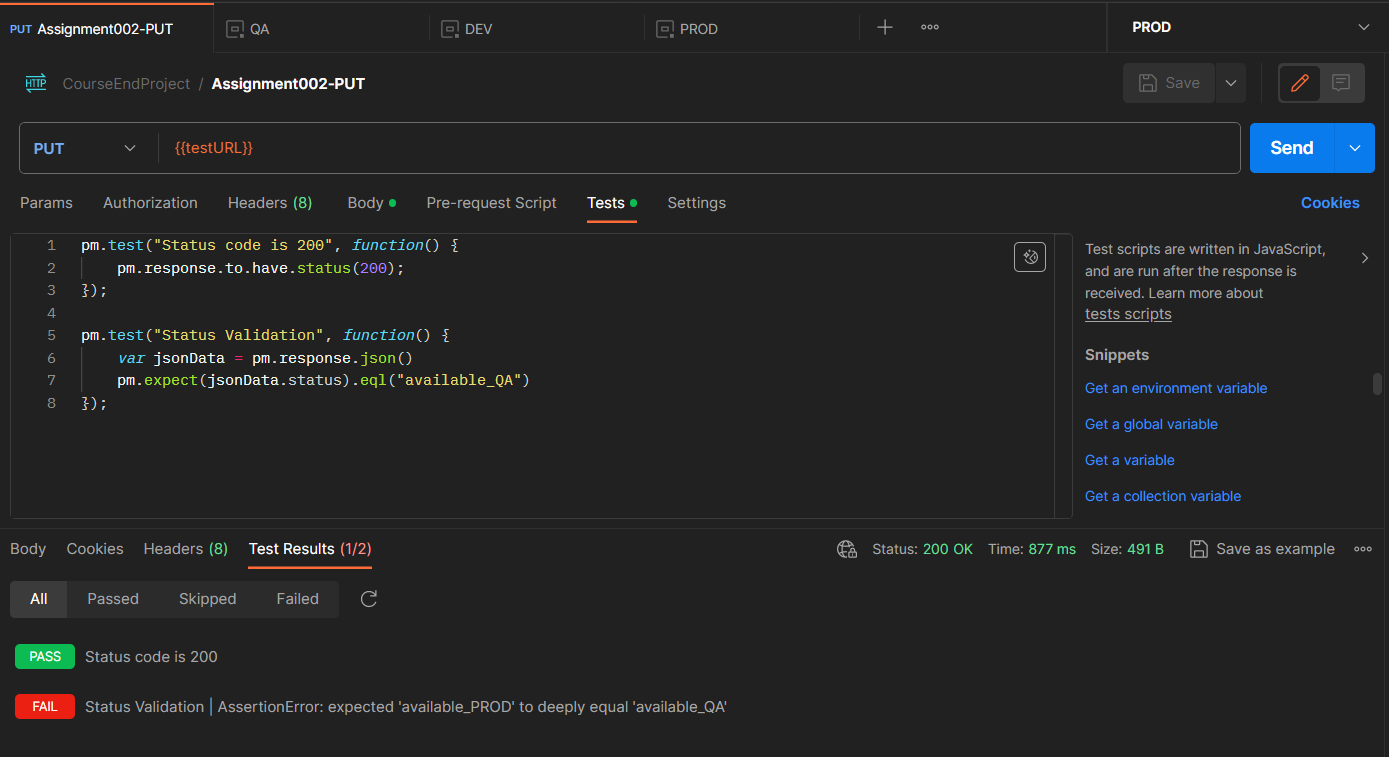
● Validate response = 200

● Validate **status** value, if it is changing as per environment in JSON response

When the Environment is QA the test is **PASS** as the value of envstatus in QA environment is “**available\_QA**”

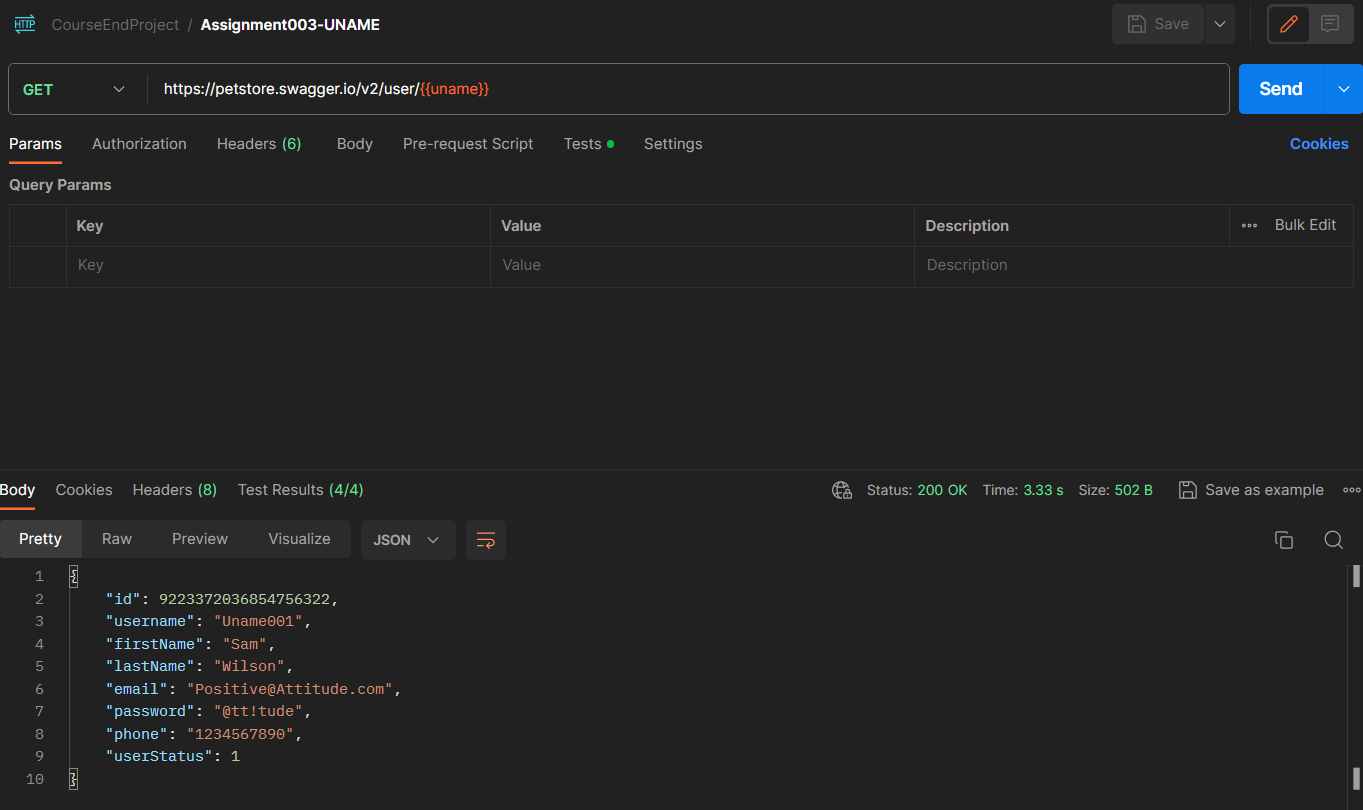


When the Environment is PROD the test is **FAIL** as the value of envstatus in PROD environment is “**available\_PROD**”

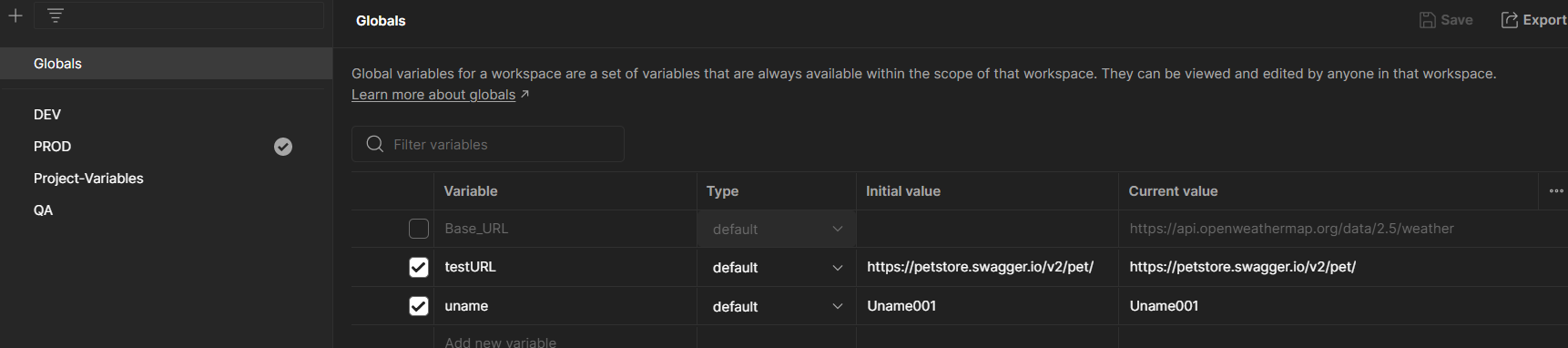


**POSTMAN ASSIGNMENT\_003:**

● Hit a **GET** call with URL: <https://petstore.swagger.io/v2/user/Uname001>



● Use Uname001 as global parameter

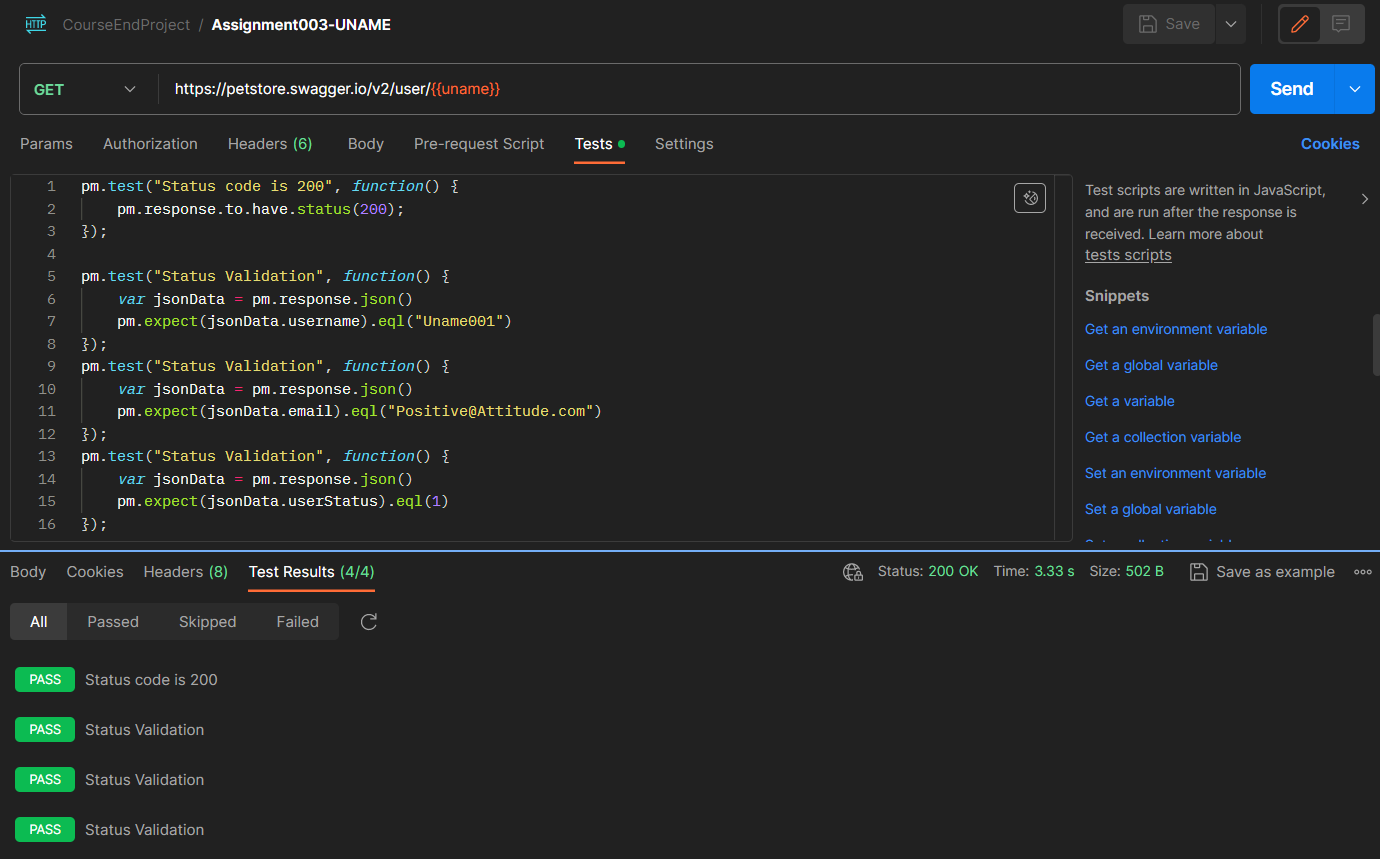


● Validate response as 200 in Postman

● Validate username = Uname001 in JSON response

● Validate email = Positive@Attitude.com in JSON response

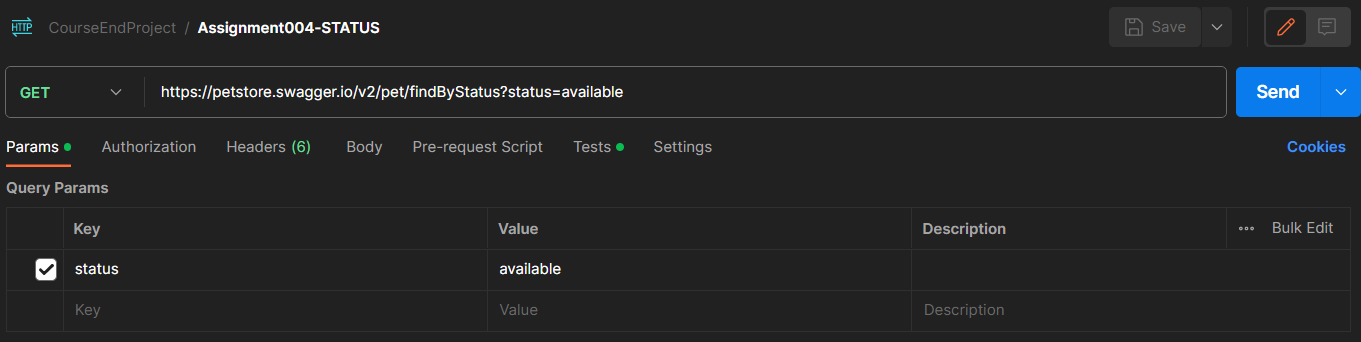
● Validate userStatus = 1 in JSON response



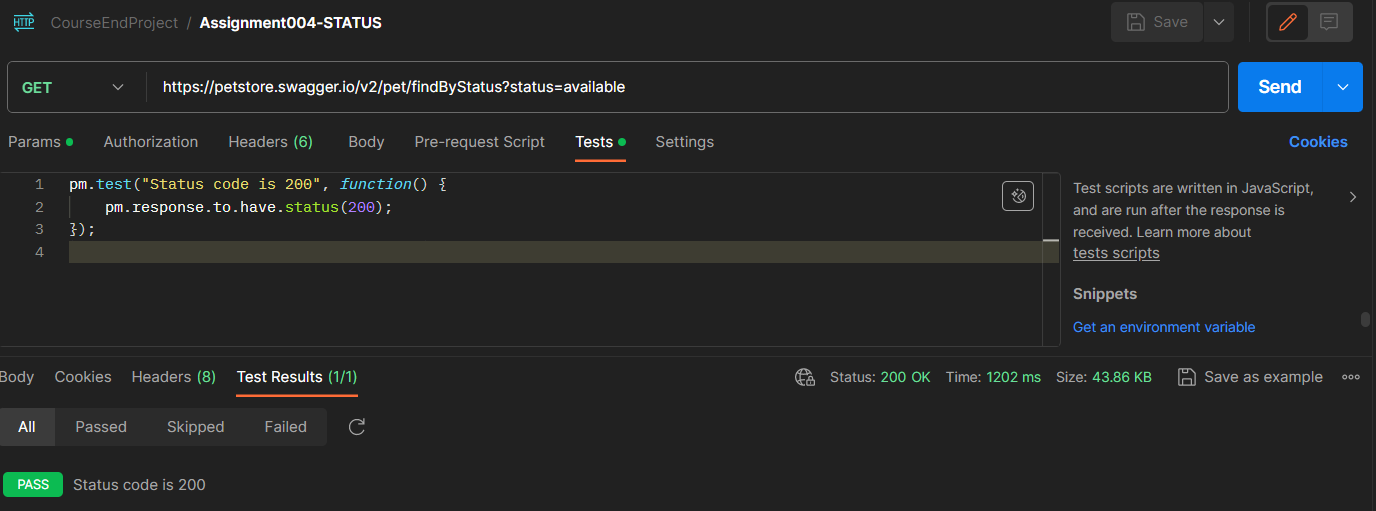
**POSTMAN ASSIGNMENT\_004:**

● Hit a **GET** call with URL: <https://petstore.swagger.io/v2/pet/findByStatus>

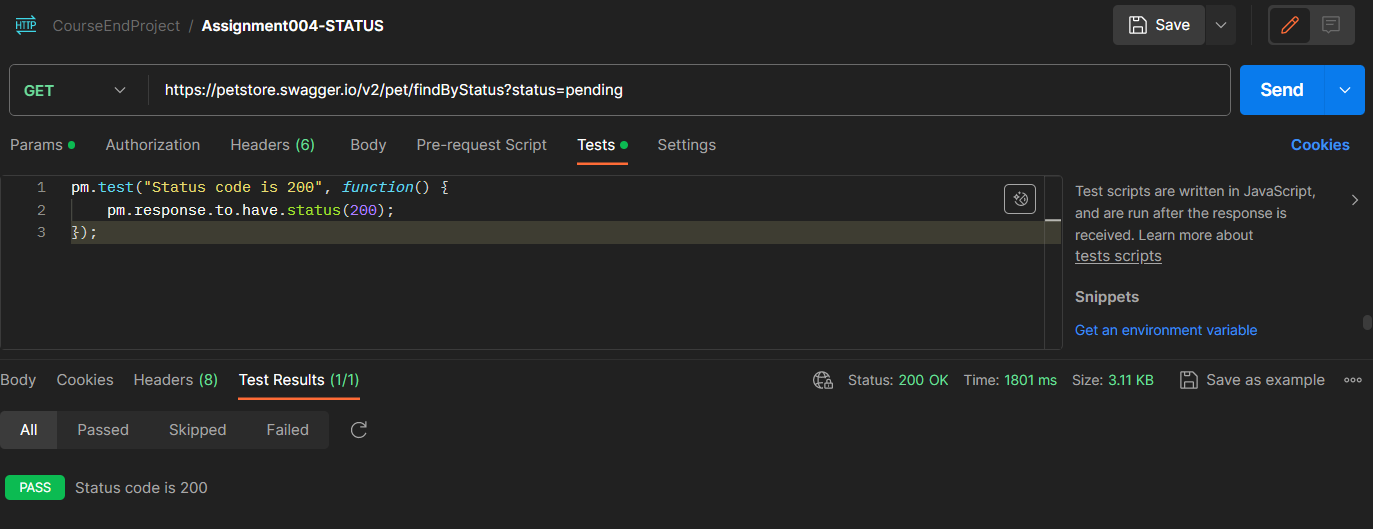
● Use Postman params as **status**



● When **status = available** and if after hitting the URL, response status = 200, then validate for all pet details that their response status = **available**

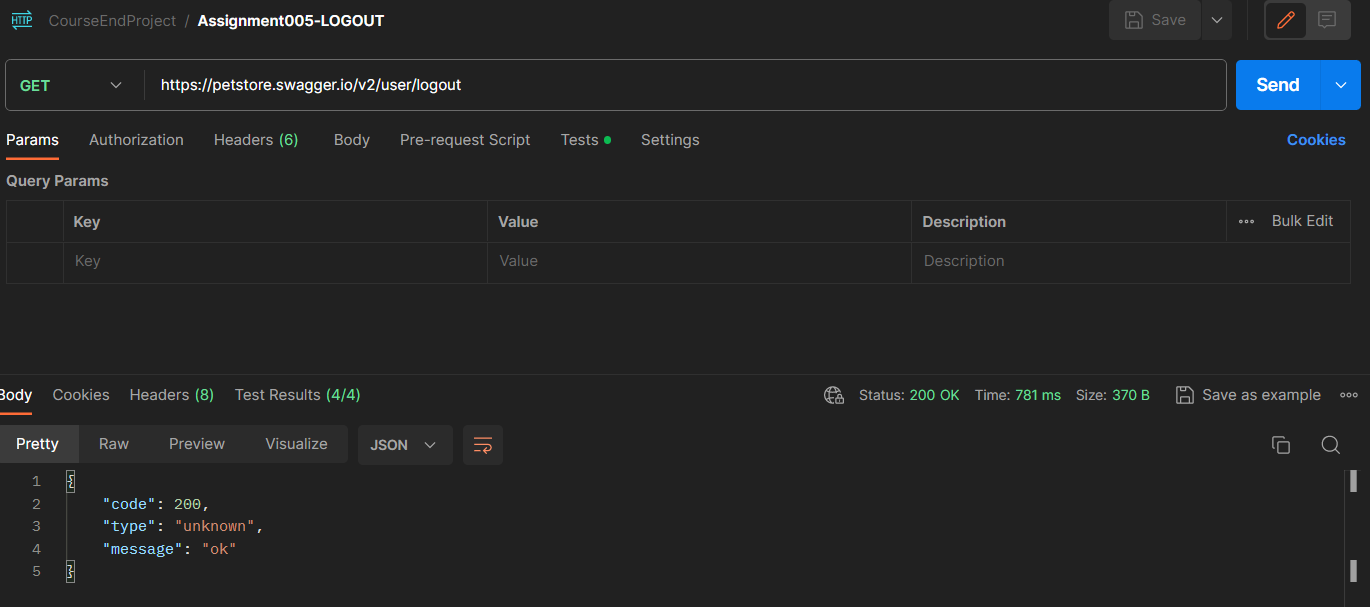


● When **status = pending** and if after hitting the URL, response status = 200, then validate for all pet details that their response status = **pending**



**POSTMAN ASSIGNMENT\_005:**

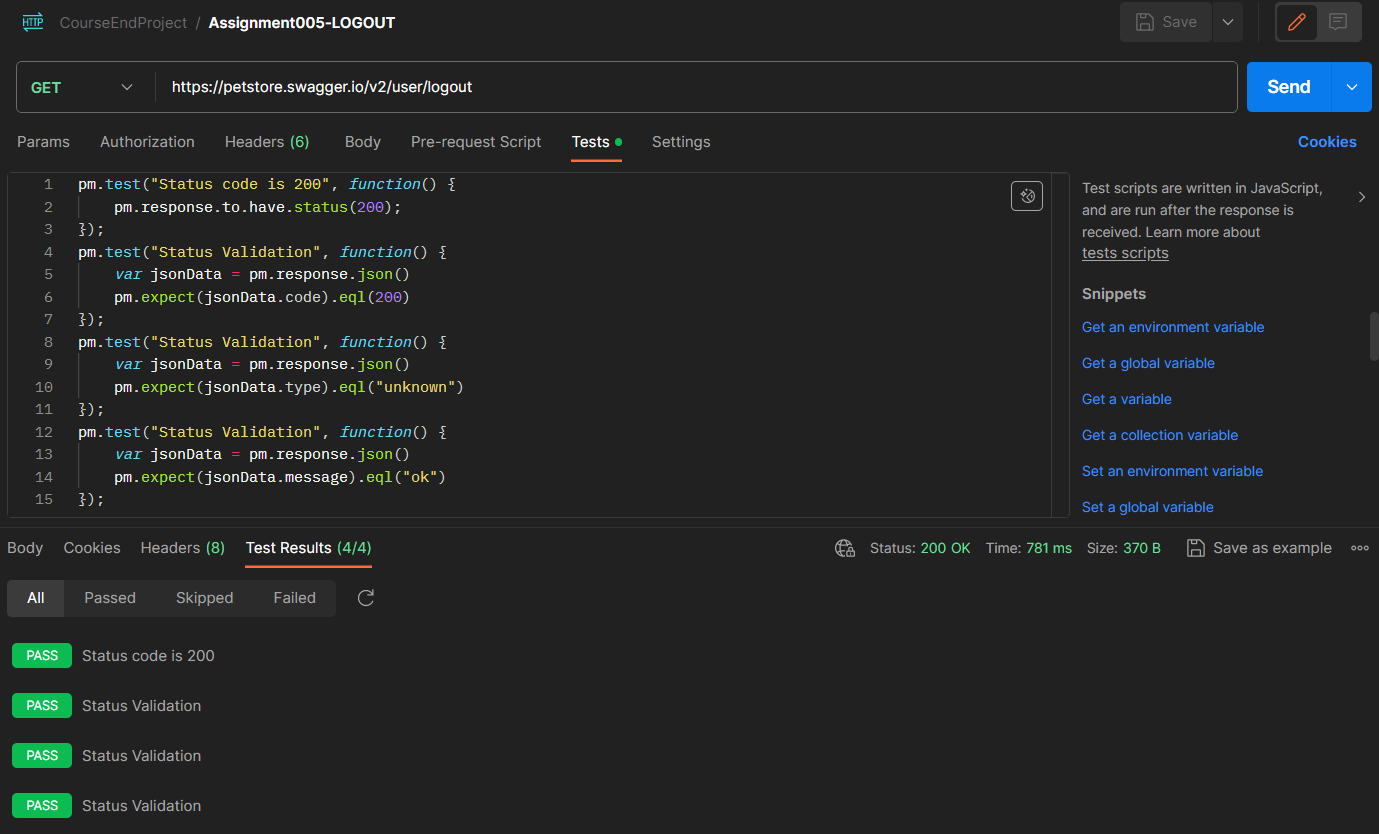
● Hit a **GET** call with URL: <https://petstore.swagger.io/v2/user/logout>



● Validate response as 200 in Postman

● Validate code = 200 in response

● Validate message = OK in response



**REST ASSURED ASSIGNMENT 001:**

● Open Eclipse/IntelliJ

● Create a Maven project

● Add Maven dependency for TestNG and REST Assured

● Create a TestNG test with below REST API execution details -

● POST CALL ○ URL: <https://petstore.swagger.io/v2/pet>

○ Service Type: POST

○ JSON Body:

{

"id": 344,

"category": {

"id": 0,

"name": "string"

},

"name": "Doggie",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available"

}

○ PetID is parameterized

○ Once POST call is successful - validate response code

○ Validate that PetID from response code should be same as request PetID

● GET CALL ○ URL: https://petstore.swagger.io/v2/pet/2003

○ Service Type: GET

○ Service URL pet **id**, which is highlighted in YELLOW color should be parameterized and be the same as POST call PetID

○ Validate response code as 200

○ Validate that response JSON body contains keys **status** and **id**

○ Validate **status** value is **available**

● DELETE CALL ○ URL: https://petstore.swagger.io/v2/pet/2003

○ Service Type: DELETE

○ Service URL pet **id**, which is highlighted in YELLOW color, should be parameterized and the same as POST call PetID

○ Validate response code is 200

○ Validate that response JSON body contains keys **code** and **message**

○ Validate **message** value is **PetID**

● POST→GET→DELETE services are interrelated. If the POST call fails, then the GET and DELETE call will also fail.

● Note: This TestNG test should run from testNG.xml file

● You can use either TestNG assertion or Hamcrest assertion

**Runner.xml**

<suite name="RestAssure Testing" verbose="1">

<test name="Test Assignment001">

<classes>

<class name="courseEndProject.Assignment001"></class>

<methods>

<include name="assignment001Post"></include>

<include name="assignment001Get"></include>

<include name="assignment001Delete"></include>

</methods>

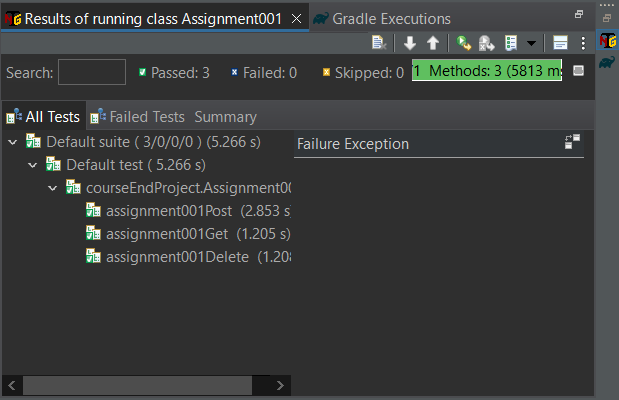
</classes>

</test>

</suite>

**OUTPUT AND SCREENSHOTS:**

**TESTNG WINDOW:**



[RemoteTestNG] detected TestNG version 7.7.1

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".

SLF4J: Defaulting to no-operation (NOP) logger implementation

SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

[INFO ] 2023-11-10 21:20:54.956 [main] Assignment001 - Course End project - Assignment001 - POST request

HTTP/1.1 200 OK

Date: Fri, 10 Nov 2023 15:50:57 GMT

Content-Type: application/json

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods: GET, POST, DELETE, PUT

Access-Control-Allow-Headers: Content-Type, api\_key, Authorization

Server: Jetty(9.2.9.v20150224)

{

"id": 645,

"category": {

"id": 0,

"name": "string"

},

"name": "duck",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available"

}

645

[INFO ] 2023-11-10 21:20:57.813 [main] Assignment001 - Course End project - Assignment001 - GET request

HTTP/1.1 200 OK

Date: Fri, 10 Nov 2023 15:50:58 GMT

Content-Type: application/json

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods: GET, POST, DELETE, PUT

Access-Control-Allow-Headers: Content-Type, api\_key, Authorization

Server: Jetty(9.2.9.v20150224)

{

"id": 645,

"category": {

"id": 0,

"name": "string"

},

"name": "duck",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available"

}

0

[INFO ] 2023-11-10 21:20:59.023 [main] Assignment001 - Course End project - Assignment001 - Delete request

HTTP/1.1 200 OK

Date: Fri, 10 Nov 2023 15:51:00 GMT

Content-Type: application/json

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods: GET, POST, DELETE, PUT

Access-Control-Allow-Headers: Content-Type, api\_key, Authorization

Server: Jetty(9.2.9.v20150224)

{

"code": 200,

"type": "unknown",

"message": "645"

}

PASSED: courseEndProject.Assignment001.assignment001Post

PASSED: courseEndProject.Assignment001.assignment001Delete

PASSED: courseEndProject.Assignment001.assignment001Get

===============================================

Default test

Tests run: 3, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 3, Passes: 3, Failures: 0, Skips: 0

===============================================

**REST ASSURED ASSIGNMENT\_002:**

● Hit a PUT call having service URL = <https://petstore.swagger.io/v2/pet>

● PUT Call request JSON body:

{ "id": 9223372016900013000, "category": {

"id": 20021, "name": "string"}, "name": "doggie", "photoUrls": [ "string" ], "tags": [ { "id": 0, "name": "string" }], "status": "available\_QA"

}

● Create three test environments as DEV, QA, PROD, where PUT call JSON body status field should be parameterized and its value should change as per environment: ○ When Environment is DEV then "status": "available\_DEV"

○ When Environment is QA then "status": "available\_QA"

○ When Environment is PROD then "status": "available\_PROD"

● All assertion/validation should be done using Hamcrest.

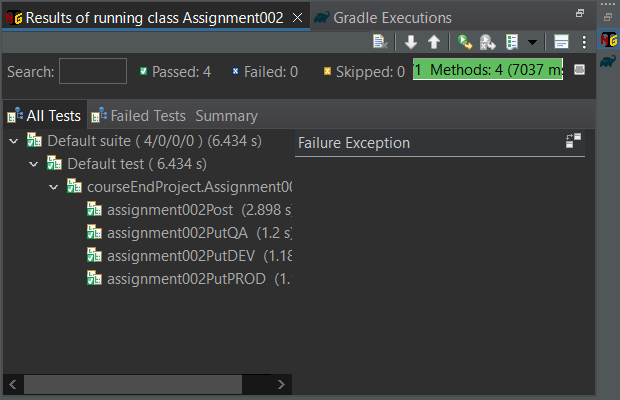
● Validate response as 200

● Validate id = 20021 in response

● Validate **status** value is changing as per environment in JSON response

**SCREENSHOTS AND OUTPUT:**

**TESTNG TEST WINDOW:**



**OUTPUT:**

[RemoteTestNG] detected TestNG version 7.7.1

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".

SLF4J: Defaulting to no-operation (NOP) logger implementation

SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

[INFO ] 2023-11-11 00:02:31.347 [main] Assignment002 - Course End project - Assignment002 - POST request

HTTP/1.1 200 OK

Date: Fri, 10 Nov 2023 18:32:33 GMT

Content-Type: application/json

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods: GET, POST, DELETE, PUT

Access-Control-Allow-Headers: Content-Type, api\_key, Authorization

Server: Jetty(9.2.9.v20150224)

{

"id": 645,

"category": {

"id": 0,

"name": "string"

},

"name": "duck",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available"

}

645

HTTP/1.1 200 OK

Date: Fri, 10 Nov 2023 18:32:35 GMT

Content-Type: application/json

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods: GET, POST, DELETE, PUT

Access-Control-Allow-Headers: Content-Type, api\_key, Authorization

Server: Jetty(9.2.9.v20150224)

{

"id": 20021,

"category": {

"id": 0,

"name": "string"

},

"name": "dogiee",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available\_QA"

}

20021

HTTP/1.1 200 OK

Date: Fri, 10 Nov 2023 18:32:36 GMT

Content-Type: application/json

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods: GET, POST, DELETE, PUT

Access-Control-Allow-Headers: Content-Type, api\_key, Authorization

Server: Jetty(9.2.9.v20150224)

{

"id": 20022,

"category": {

"id": 0,

"name": "string"

},

"name": "dog",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available\_DEV"

}

20022

HTTP/1.1 200 OK

Date: Fri, 10 Nov 2023 18:32:37 GMT

Content-Type: application/json

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods: GET, POST, DELETE, PUT

Access-Control-Allow-Headers: Content-Type, api\_key, Authorization

Server: Jetty(9.2.9.v20150224)

{

"id": 20023,

"category": {

"id": 0,

"name": "string"

},

"name": "cat",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available\_PROD"

}

20023

PASSED: courseEndProject.Assignment002.assignment002PutQA

PASSED: courseEndProject.Assignment002.assignment002PutDEV

PASSED: courseEndProject.Assignment002.assignment002Post

PASSED: courseEndProject.Assignment002.assignment002PutPROD

===============================================

Default test

Tests run: 4, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 4, Passes: 4, Failures: 0, Skips: 0

===============================================

**REST ASSURED ASSIGNMENT\_003:**

● Create a TestNG test and implement the scenario

● Hit a GET call with URL: <https://petstore.swagger.io/v2/user/Uname001>

● All assertion/validation should be done using Hamcrest

● Validate response as 200

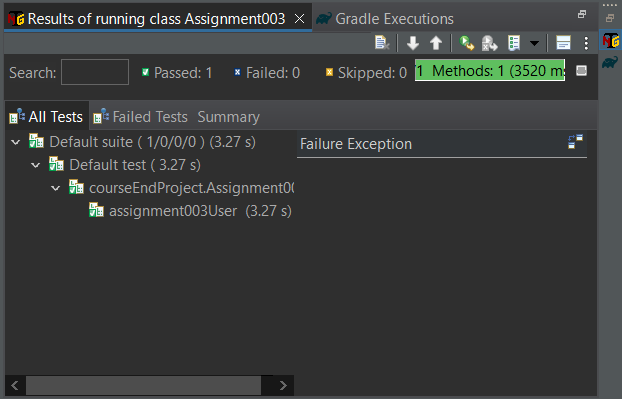
● Validate username = Uname001 in JSON response

● Validate email = Positive@Attitude.com in JSON response

● Validate userStatus = 1 in JSON response

**OUTPUT AND SCREENSHOTS:**

**TESTNG TEST WINDOW**



**OUTPUT:**

[RemoteTestNG] detected TestNG version 7.7.1

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".

SLF4J: Defaulting to no-operation (NOP) logger implementation

SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

HTTP/1.1 200 OK

Date: Fri, 10 Nov 2023 19:09:07 GMT

Content-Type: application/json

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods: GET, POST, DELETE, PUT

Access-Control-Allow-Headers: Content-Type, api\_key, Authorization

Server: Jetty(9.2.9.v20150224)

{

"id": 9223372036854775807,

"username": "Uname001",

"firstName": "Uname001",

"lastName": "Uname001",

"email": "Positive@Attitude.com",

"password": "string",

"phone": "string",

"userStatus": 1

}

PASSED: courseEndProject.Assignment003.assignment003User

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

===============================================

**REST ASSURED ASSIGNMENT\_004:**

● Create a TestNG test and implement the below scenario

● Hit a **GET** call with URL: https://petstore.swagger.io/v2/user/login

● Use REST Assured basic authentication with **username** = Uname001 and **password** = @tt!tude

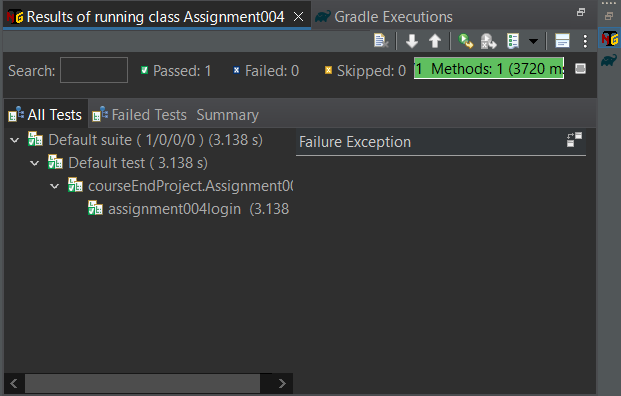
● All assertion/validation should be done using Hamcrest

● Validate code = 200 in response

● Validate response message contains text ‘logged in user session’

**OUTPUT AND SCREENSHOTS:**

**TESTNG TEST WINDOW:**



**OUTPUT:**

[RemoteTestNG] detected TestNG version 7.7.1

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".

SLF4J: Defaulting to no-operation (NOP) logger implementation

SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

[INFO ] 2023-11-11 10:27:32.326 [main] Assignment004 - Course End project - Assignment003 - GET login request

HTTP/1.1 200 OK

Date: Sat, 11 Nov 2023 04:57:33 GMT

Content-Type: application/json

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods: GET, POST, DELETE, PUT

Access-Control-Allow-Headers: Content-Type, api\_key, Authorization

X-Expires-After: Sat Nov 11 05:57:33 UTC 2023

X-Rate-Limit: 5000

Server: Jetty(9.2.9.v20150224)

{

"code": 200,

"type": "unknown",

"message": "logged in user session:1699678653242"

}

PASSED: courseEndProject.Assignment004.assignment004login

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

===============================================

**REST ASSURED ASSIGNMENT\_005:**

● Create a TestNG test

● Hit a GET call with URL: <https://petstore.swagger.io/v2/pet/findByStatus>

● Use REST Assured params as status

● All assertion/validation should be done using Hamcrest

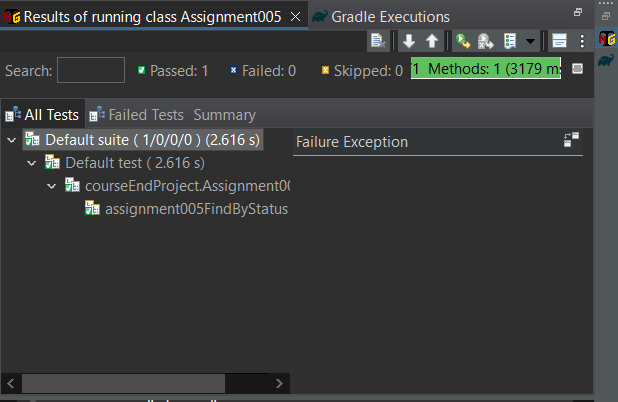
● When status = available and if after hitting the URL, response status = 200, then validate for all pet details that their response status = available

● When status = pending and if after hitting the URL, response status = 200, then validate for all pet details that their response status = pending

● When status = sold and if after hitting the URL, response status = 200, then validate for all pet details that their response status = sold

**OUTPUT AND SCREENSHOTS:**

**TESTNG TEST WINDOW:**



**OUTPUT:**

[RemoteTestNG] detected TestNG version 7.7.1

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".

SLF4J: Defaulting to no-operation (NOP) logger implementation

SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

[INFO ] 2023-11-11 10:44:11.653 [main] Assignment005 - Course End project - Assignment005 - GET status request

HTTP/1.1 200 OK

Date: Sat, 11 Nov 2023 05:14:13 GMT

Content-Type: application/json

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods: GET, POST, DELETE, PUT

Access-Control-Allow-Headers: Content-Type, api\_key, Authorization

Server: Jetty(9.2.9.v20150224)

[

{

"id": 9223372016900013798,

"category": {

"id": 82641878,

"name": "dolor dolore nisi Duis"

},

"name": "doggie",

"photoUrls": [

"officia in in enim",

"eu sed"

],

"tags": [

{

"id": 83694122,

"name": "esse sit in aute anim"

},

{

"id": -91142734,

"name": "sit deserunt mollit ea"

}

],

"status": "pending"

},

{

"id": 9223372016900013816,

"category": {

"id": 82212339,

"name": "sed nulla minim"

},

"name": "doggie",

"photoUrls": [

"veniam officia",

"amet Ut Excepteur irure"

],

"tags": [

{

"id": -64599653,

"name": "nisi qui"

},

{

"id": -20636986,

"name": "et est"

}

],

"status": "pending"

}

]

PASSED: courseEndProject.Assignment005.assignment005FindByStatus

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

===============================================

**REST ASSURED ASSIGNMENT\_006:**

● Create a TestNG test

● Hit a GET call with URL: <https://petstore.swagger.io/v2/user/logout>

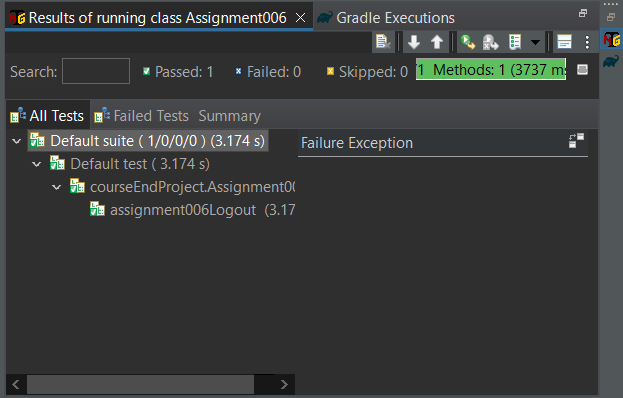
● All assertion/validation should be done using Hamcrest

● Validate response as 200

● Validate code = 200 in response

● Validate message = OK in response

**TESTNG TEST WINDOW:**



**OUTPUT:**

[RemoteTestNG] detected TestNG version 7.7.1

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".

SLF4J: Defaulting to no-operation (NOP) logger implementation

SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

[INFO ] 2023-11-11 11:20:13.881 [main] Assignment006 - Course End project - Assignment006 - GET logout request

HTTP/1.1 200 OK

Date: Sat, 11 Nov 2023 05:50:14 GMT

Content-Type: application/json

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods: GET, POST, DELETE, PUT

Access-Control-Allow-Headers: Content-Type, api\_key, Authorization

Server: Jetty(9.2.9.v20150224)

{

"code": 200,

"type": "unknown",

"message": "ok"

}

PASSED: courseEndProject.Assignment006.assignment006Logout

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

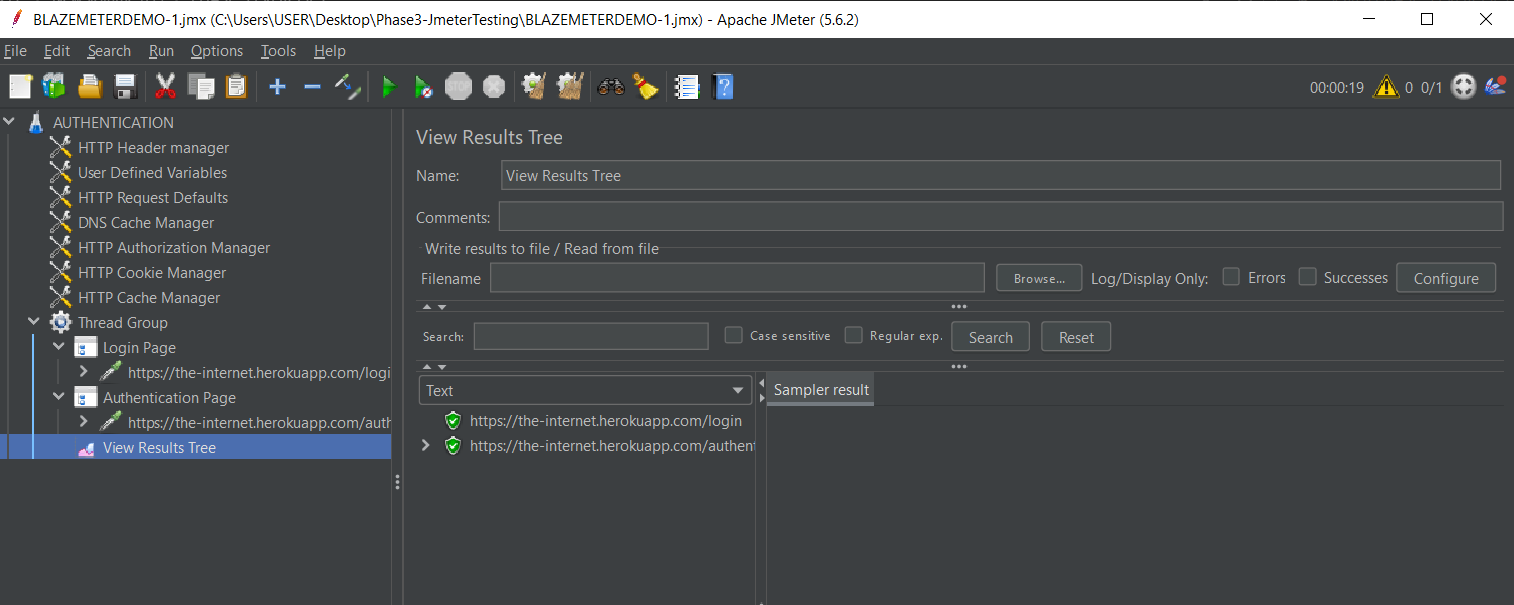
===============================================

**JMETER ASSIGNMENT:**

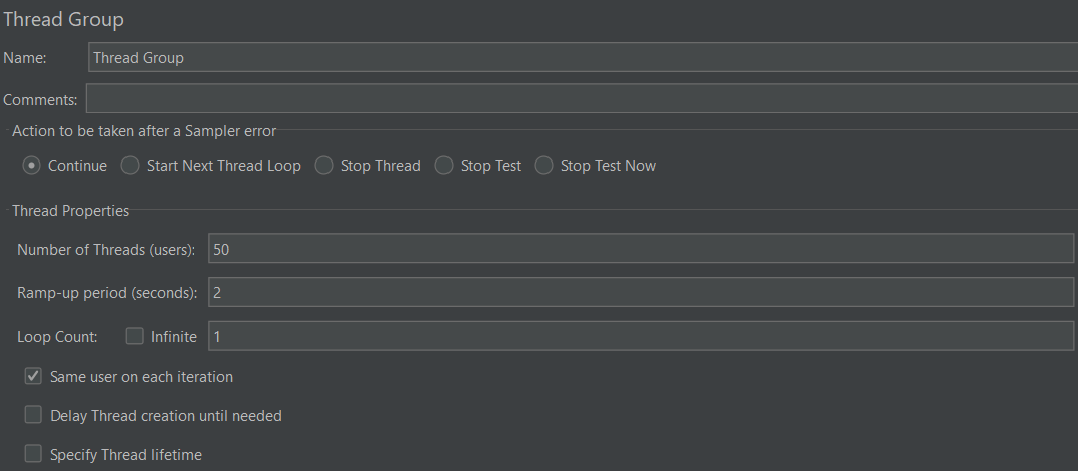
● Open JMeter

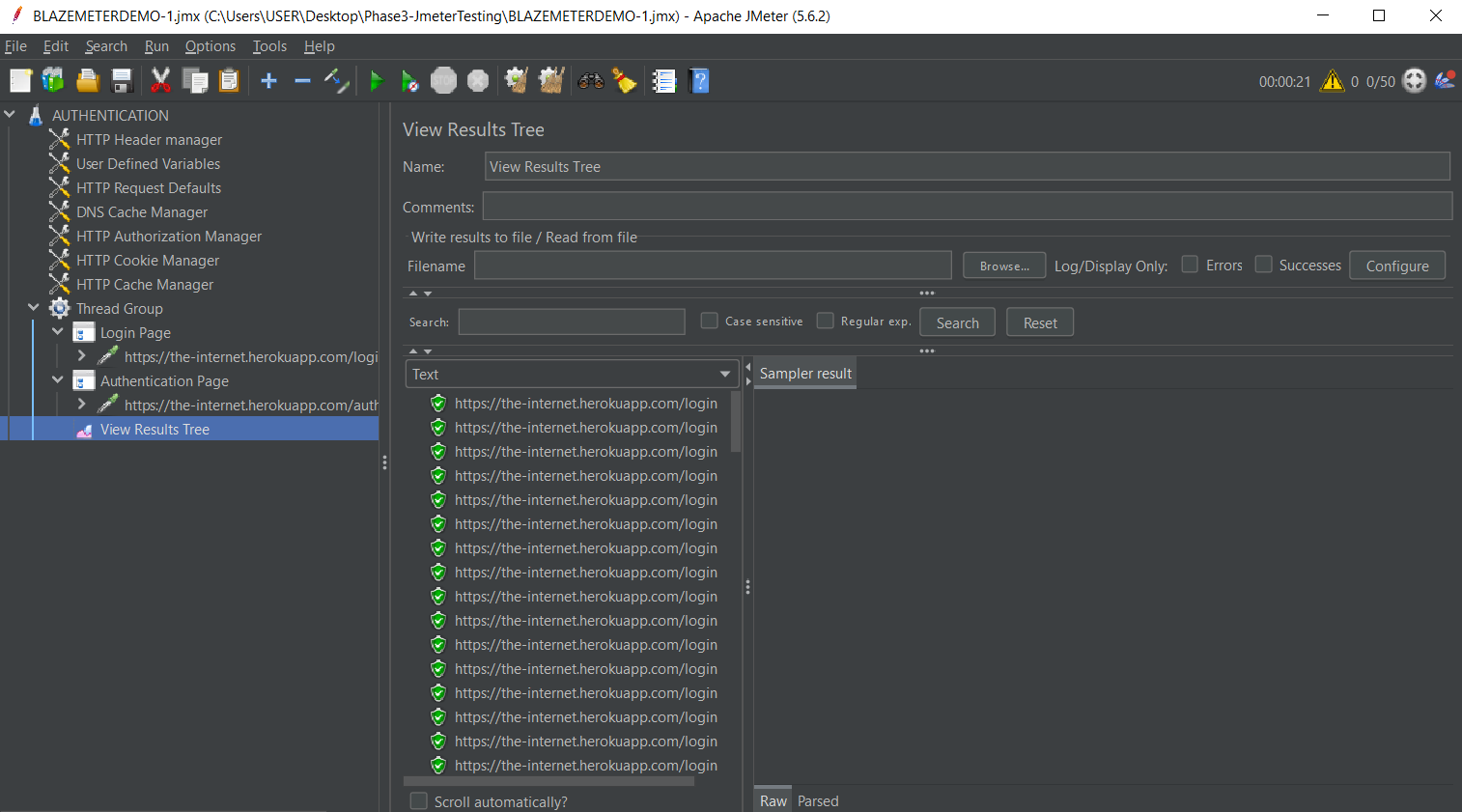
● Validate <https://httpbin.org/basic-auth/user/passwd> link using HTTP Authentication manager

● Default username is user, and the default password is passwd

****

● Increase the thread count and do lead testing

****

****