**Read Me:**

**Version:1**

KABIRE2ERESTAPIFRAMEWORK is an End-to-End Data driven test framework with test cases written in excel files. The test framework is designed in such a way that the framework can be easily used by cross functional team to automate end to end/integration test cases. Here multiple API calls are grouped together to execute sequentially to validate certain E2E functionality. In each API call we validate the response against expectations and at the end of the cycle it will tell us which API validation failed. While designing the framework one of the aspects that was in my mind was to make this framework as loose coupling as possible so cross functional implementation team/teams can change this according to their team needs.

**Version:2**

I am still working on the project when time permits and adding the following features in version 2:

1. Update test execution report with Allure Report
2. More Configuration from testNg.xml file
3. Added log4j logging support for better logging and debugging.
4. Updated validation method.
5. Adding better configuration files to keep various data separate.
6. Adding support to run the test cases with testNG.xml files as well as Jenkins Job run.

**How to Run the test Case:**

**Right** click on *DataDrivenTestingUsingExcelFile.java* ->**Run as->TestNG Test.** It will run the test cases and show the test result like below:

**Test Run Result:**

A screenshot of a computer

Description automatically generated

**emailable-report.html:**

A screenshot of a computer

Description automatically generated

**Default test.html report:**

A green screen shot of a computer

Description automatically generated

**Console Output:**

[RemoteTestNG] detected TestNG version 7.4.0

Sequence ID: 1

Api Method Name ID: postbooking

Executing : postbooking API

My booking id from Response:::1027

DataBean Map Data:::1027

DataBean First Name::Van

DataBean Last Name::Dam

Sequence ID: 2

Api Method Name ID: getbooking

Executing: GetBooking API

The Array Value here:[Ljava.lang.String;@58ae402b

Returning Some String Value The lastname found in the response:!!!! Dam

JSONPath is: $.lastname, Query: EQUALS, Base Value: "Dam", Actual Value: "Dam"

The Array Value here:[Ljava.lang.String;@2668c286

Returning Some String Value The firstname found in the response:!!!! Van

JSONPath is: $.firstname, Query: EQUALS, Base Value: "Van", Actual Value: "Van"

Sequence ID: 3

Api Method Name ID: gettoken

Executing: GetToken API

Token Id : bcd42591c8dee1d

The Array Value here:[Ljava.lang.String;@53f0d09c

Returning Some String Value The token found in the response:!!!! bcd42591c8dee1d

JSONPath is: $.token, Query: KEYEXISTS, Base Value: TRUE, Actual Value: true

Sequence ID: 4

Api Method Name ID: putbooking

Executing: PutBooking API

The Array Value here:[Ljava.lang.String;@83bb0f

Returning Some String Value The additionalneeds found in the response:!!!! Rib Eye Steak

JSONPath is: $.additionalneeds, Query: EQUALS, Base Value: "Rib Eye Steak", Actual Value: "Rib Eye Steak"

Sequence ID: 5

Api Method Name ID: getbooking

Executing: GetBooking API

The Array Value here:[Ljava.lang.String;@706ceca6

Returning Some String Value The additionalneeds found in the response:!!!! Rib Eye Steak

JSONPath is: $.additionalneeds, Query: EQUALS, Base Value: "Rib Eye Steak", Actual Value: "Rib Eye Steak"

Sequence ID: 6

Api Method Name ID: patchbooking

Executing: PatchBooking API

The Array Value here:[Ljava.lang.String;@3fbe503c

Returning Some String Value The lastname found in the response:!!!! Kabir

JSONPath is: $.lastname, Query: EQUALS, Base Value: "Kabir", Actual Value: "Kabir"

Sequence ID: 7

Api Method Name ID: getbooking

Executing: GetBooking API

The Array Value here:[Ljava.lang.String;@7b2bf745

Returning Some String Value The lastname found in the response:!!!! Kabir

JSONPath is: $.lastname, Query: EQUALS, Base Value: "Kabir", Actual Value: "Kabir"

The Array Value here:[Ljava.lang.String;@6fe595dc

Returning Some String Value The additionalneeds found in the response:!!!! Rib Eye Steak

JSONPath is: $.additionalneeds, Query: EQUALS, Base Value: "Rib Eye Steak", Actual Value: "Rib Eye Steak"

Sequence ID: 8

Api Method Name ID: deletebooking

Executing: DeleteBooking API

PASSED: DataDrivenTesting({APIMethodName=GetToken, BaseLineStatusCode=200, InputData={

"username": "admin",

"password": "password123"

}, SequenceID=3, Test\_Case\_Name=FlightBookingE2E, ValidationContent=[$.token],[KEYEXISTS],[TRUE],[1]})

PASSED: DataDrivenTesting({APIMethodName=PatchBooking, BaseLineStatusCode=200, InputData={

"lastname": "Kabir"

}, SequenceID=6, Test\_Case\_Name=FlightBookingE2E, ValidationContent=[$.lastname],[EQUALS],[Kabir]})

PASSED: DataDrivenTesting({APIMethodName=GetBooking, BaseLineStatusCode=200, InputData=, SequenceID=5, Test\_Case\_Name=FlightBookingE2E, ValidationContent=[$.additionalneeds],[EQUALS],[Rib Eye Steak]})

PASSED: DataDrivenTesting({APIMethodName=GetBooking, BaseLineStatusCode=200, InputData=, SequenceID=2, Test\_Case\_Name=FlightBookingE2E, ValidationContent=[$.lastname],[EQUALS],[Dam]

[$.firstname],[EQUALS],[Van]})

PASSED: DataDrivenTesting({APIMethodName=GetBooking, BaseLineStatusCode=200, InputData=, SequenceID=7, Test\_Case\_Name=FlightBookingE2E, ValidationContent=[$.lastname],[EQUALS],[Kabir]

[$.additionalneeds],[EQUALS],[Rib Eye Steak]})

PASSED: DataDrivenTesting({APIMethodName=PutBooking, BaseLineStatusCode=200, InputData={

"firstname": "Van",

"lastname": "Dam",

"totalprice": 1000,

"depositpaid": true,

"bookingdates": {

"checkin": "2018-01-01",

"checkout": "2019-01-01"

},

"additionalneeds": "Rib Eye Steak"

}, SequenceID=4, Test\_Case\_Name=FlightBookingE2E, ValidationContent=[$.additionalneeds],[EQUALS],[Rib Eye Steak]})

PASSED: DataDrivenTesting({APIMethodName=DeleteBooking, BaseLineStatusCode=201, InputData=, SequenceID=8, Test\_Case\_Name=FlightBookingE2E, ValidationContent=})

PASSED: DataDrivenTesting({APIMethodName=PostBooking, BaseLineStatusCode=201, InputData={

"firstname": "Van",

"lastname": "Dam",

"totalprice": 1000,

"depositpaid": true,

"bookingdates": {

"checkin": "2018-01-01",

"checkout": "2019-01-01"

},

"additionalneeds": "Muslim Food"

}, SequenceID=1, Test\_Case\_Name=FlightBookingE2E, ValidationContent=})

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

Default test

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

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

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

Default suite

Total tests run: 8, Passes: 8, Failures: 0, Skips: 0

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