# - QA assignment -

Amina Poplata

October 21, 2020.

# 1 The assignment

Investigate Skyscanner API service and create test plan: `https://rapidapi.com/skyscanner/api/skyscanner-flight-search/details` :

1. Write test cases for this service (concentrate on real user flows, potentially combining multiple api calls in one test case)
2. From all the test cases, identify what you think represents a Smoke Test
3. Identify positive and negative test cases
4. With a tool or programming/scripting language (JMeter, Python, Ruby, Java, Javascript...), automate the smoke test from #2 (it is plus if you are using some testing framework in any of the listed programming/scripting languages)
5. Create Github repository and push your assignment solution with the documentation on how to run the automated tests there
6. Report bugs if you find any, with detailed steps to reproduce.

# 2 Test cases

Test cases for this service along with the matching test scenarios and details about test cases are written in the attached *.xlsx* file in *GitHub* repository named *SkyscannerAPI_Scenario.xlsx*.

# 3 Smoke Test

Smoke test represents a test using which the core functions of the API are tested. The core functionality of the tested API would be the *Browse Flight Prices* functionality, so all the test scenarios and test cases which are testing this functionality can be represented as smoke testing (for example: testing *Browse Quotes* scenario (scenario #2) and matching test cases #1, #2, #3, #4 and #8 from the *SkyscannerAPI_Scenario.xlsx* file).

# 4 Positive and negative test cases

Positive test cases are test cases in which valid input data is used for testing. Example of positive test cases from the written test scenarios and test cases in *SkyscannerAPI_Scenario.xlsx* file are test cases:

- Test Scenario 1: Test Case 1
- Test scenario 2: Test Case 1
- Test Scenario 2: Test Case 2
- Test Scenario 3: Test Case 1.

On the contrary, negative test cases are test cases in which invalid input data is used for testing. Example of negative test cases from the written test scenarios and test cases in *SkyscannerAPI_Scenario.xlsx* file are:

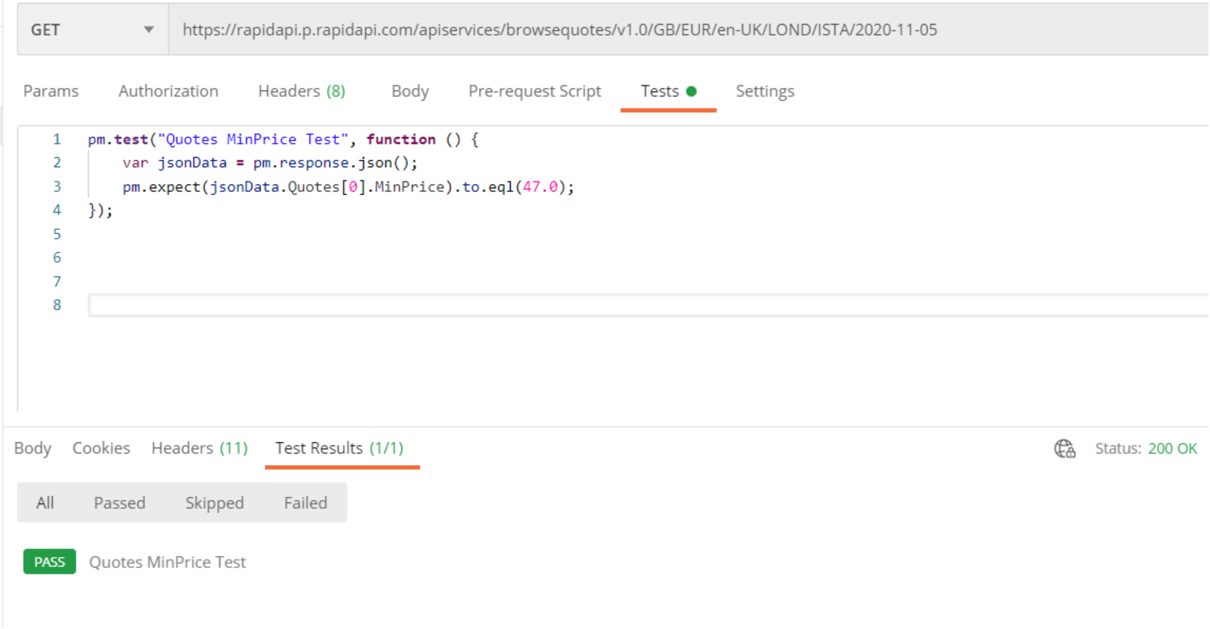
- Test Scenario 1: Test Case 5
- Test Scenario 2: Test Case 8
- Test Scenario 3: Test Case 2
- Test Scenario 3: Test Case 3
- TEst SCenario 8: Test Case 2.

# 5 Test automation

For test automation exapmle, test case 1 from test scenario 2 (in file *SkyscannerAPI_Scenario.xlsx*) was chosen. Test automation was done using *Postman* tool and also using *Python* programming language.

Considering that result information can be fluid, primary test check is if the matching *GET* request gives *non-empty* objects as result. That was done using *Python*: if the *Quotes*, *Places*, *Carriers* and *Currencies* objects aren't empty, test is considered successful. Of course, more detailed testing can be done: for example every instance of mentioned objects can be compared with the expecting result and if they are equal, test should be considered successful and if they are not equal then test failed and appropriate message can be printed. In this assignment, just a simple automation example was done for demonstration purpose. *Python* code of this test is attached on *GitHub* repository.

In *Postman*, for example, one test is made to compare *MinPrice* instance of the of *Quotes[0]* with the expected price (manually tested earlier). The result is shown in the figure 1. This is also done for demonstration only and more tests can be written to for every test case.



GET https://rapidapi.p.rapidapi.com/apiservices/browsequotes/v1.0/GB/EUR/en-UK/LOND/ISTA/2020-11-05

```
pm.test("Quotes MinPrice Test", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.Quotes[0].MinPrice).to.eql(47.0);
});
```

Status: 200 OK

PASS Quotes MinPrice Test

**Figure 1:** Test example from *Postman*

# 6 GitHub repository

GitHub repository link which contains all used files is: `https://github.com/aminnadz/Atlantbh_AminaPoplata`.

# 7 Bugs

All the test that failed are marked as *FAIL* in the *SkyscannerAPI_Scenario.xlsx* file where more details about tests can be found (input data and results of testing).