Ahmed Abokhalifa
01-29-2020

payconiq

QA Engineer assignment: API testing

Solution URL on GitHub

https://github.com/Abokhalifa/com.payconiq

Features of the solution of Gists

1. Written in Java using Rest-Assured framework
2. Design pattern like Page Object Model
3. Data driven tests from external data source
4. TestNg is used as a testing framework.
5. Basic CRUD functions of   are included.
6. Autonomous tests that do not depend on one another
7. Modular solution with Maven dependencies

Future Plan

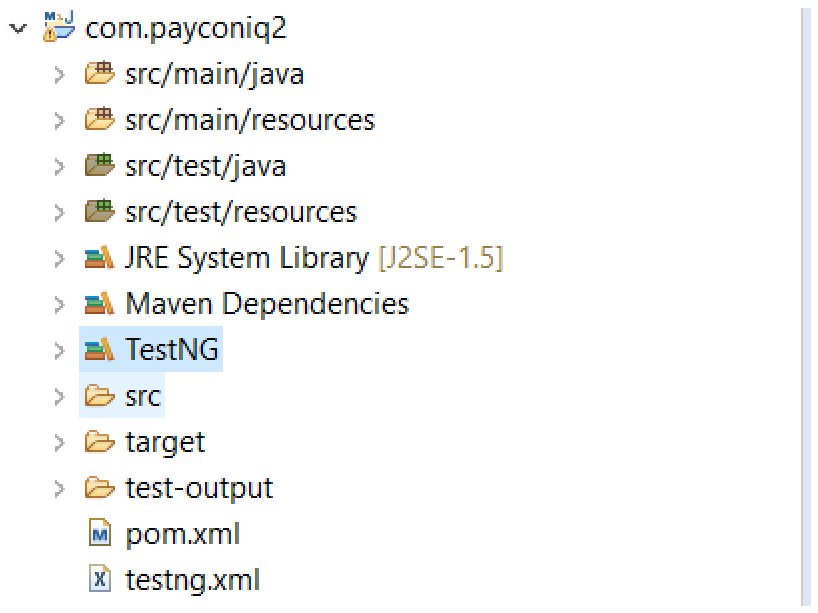
1. Cover more tests based on POISED test strategy.
2. Handle exceptions.
3. Cover Negative, Security and Performance.
4. Improve the test report.
5. Work with pagination.
6. Validate schema of the response.

- Test strategy
    - POISED API test strategy is considered.
    - A mix of unit, integration, and system end-to-end tests are included.
    - Performance → To be covered in a later sprint.
    - Security → To be covered in a later sprint.
    - Negative tests → This is crucial in API testing and will be covered in a later sprint.

- Programming skills
    - Inheritance and encapsulation are included among classes to improve reusability and extendibility.
- How you test APIs
    - Same as in Test Strategy section above
- The maintainability of the solution
    - Separation of concerns is considered to isolate responsibilities of each class and hence more easier debugging is guaranteed.
- Show good structuring of tests for reusability /naming convention
    - Clear and up to the point folder structures
    - Consistent naming convention is used from the common practice of java naming convention
    - A design pattern like the Pag Object Model is used

Instruction to run the solution

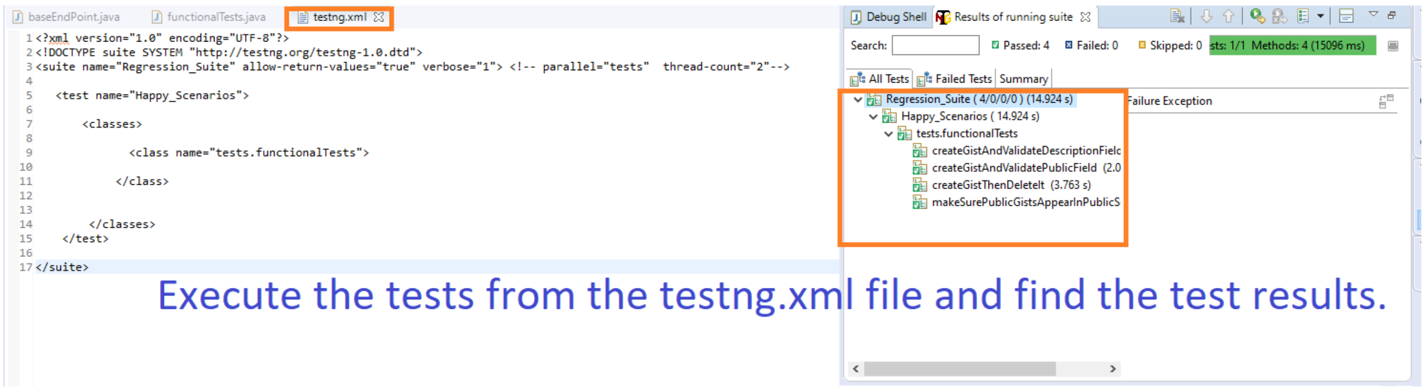
1. Make sure Maven installs the dependencies in the POM.xml file. To install the dependencies, right click the project directory → Maven → Update Project or simply Alt + F5.
2. Make sure to add TestNg framework.
3. Make sure the folder structure is something like the below image:



```
com.payconiq2
  src/main/java
  src/main/resources
  src/test/java
  src/test/resources
  JRE System Library [J2SE-1.5]
  Maven Dependencies
  TestNG
  src
  target
  test-output
  pom.xml
  testng.xml
```

Ahmed Abokhalifa
01-29-2020

4. Set the "Token" parameter @ "`baseEndPoint`" class before using the app
5. Execute the tests from testing.xml and see the results as shown below.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Regression_Suite" allow-return-values="true" verbose="1"> <!-- parallel="tests"  thread-count="2"-->

  <test name="Happy_Scenarios">

      <classes>

            <class name="tests.functionalTests">

            </class>


       </classes>
   </test>

</suite>
```

```
Passed: 4   Failed: 0   Skipped: 0  sts: 1/1  Methods: 4 (15096 ms)

Regression_Suite ( 4/0/0/0 ) (14.924 s)
  Happy_Scenarios ( 14.924 s)
    tests.functionalTests
      createGistAndValidateDescriptionFielc
      createGistAndValidatePublicField  (2.0
      createGistThenDeleteIt  (3.763 s)
      makeSurePublicGistsAppearInPublicS
```

Execute the tests from the testng.xml file and find the test results.

Ahmed Abokhalifa
01-29-2020