


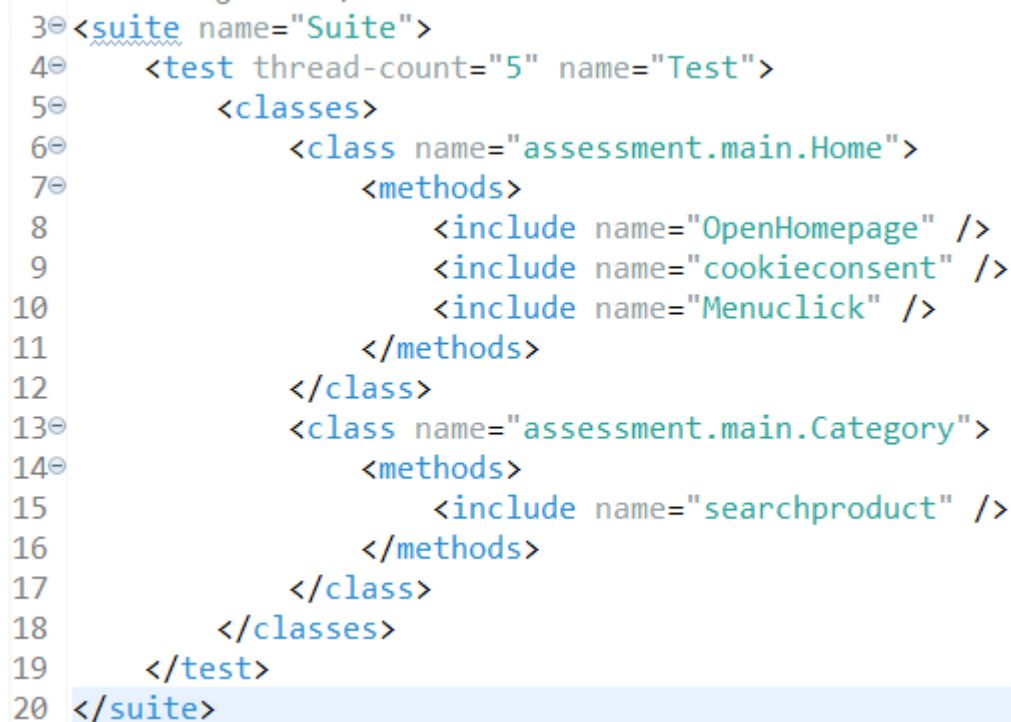
Practical Overview

1. Data file "Input.xlsx" in the data folder for input
2. Pom.xml file to set dependencies which maven can auto-download
3. Masterdata.properties is used to set the base URL according to the environment and set the browser to execute test cases on it.

A screenshot of a text editor window titled "Masterdata.properties". The file contains five lines of configuration: 1. A comment line: "#URL QA environment, pre-production environment, production environment". 2. A URL line: "URL=https://www.douglas.de/de". 3. A comment line: "#Browser :: Firefox, Chrome". 4. A browser line: "browser=Chrome". 5. An empty line.

```
1#URL QA environment, pre-production environment, production environment
2URL=https://www.douglas.de/de
3#Browser :: Firefox, Chrome
4browser=Chrome
5
```

4. Textng.xml to create a test suite

A screenshot of a text editor window showing an XML file. The XML defines a test suite named "Suite" containing a test named "Test" with 5 threads. The test includes two classes: "assessment.main.Home" and "assessment.main.Category". The "Home" class has three methods: "OpenHomepage", "cookieconsent", and "Menuclick". The "Category" class has one method: "searchproduct".

```
3<suite name="Suite">
4  <test thread-count="5" name="Test">
5    <classes>
6      <class name="assessment.main.Home">
7        <methods>
8          <include name="OpenHomepage" />
9          <include name="cookieconsent" />
10         <include name="Menuclick" />
11        </methods>
12      </class>
13      <class name="assessment.main.Category">
14        <methods>
15          <include name="searchproduct" />
16        </methods>
17      </class>
18    </classes>
19  </test>
20</suite>
```

- a. First, it will open the home page and validate the title
 - b. Then accept cookie consent
 - c. It will click on the "Perfume" menu after accepting cookie consent
 - d. On category page, it will search products as per search criteria (will take data from input.xlsx file)
5. Utility class, which is in assessment.utility package contains common methods
 6. Homepage and categorypage classes that are in assessment.pages package contains the path of different elements of respective pages

7. Base class of assessment.the main package contains before and after methods, which will be executed before and after suite execution to initialize the driver, collect input data, and collect information from the properties file
8. test-output folder contains result of execution, open index.html file to view result

The screenshot displays the TestNG test results interface. At the top, a dark green header bar contains the text "1 suite" on the left, a "Switch Retro Theme" button in the center, and "Test results" on the right. Below the header, the interface is divided into two main sections. The left section, titled "All suites" with a dropdown arrow, contains a "Suite" sub-section. Under "Suite", there is an "Info" section showing the test file path "D:\Workspace\Java\Practical_Assessment\testng.xml", "1 test", "0 groups", "Times", "Reporter output", "Ignored methods", and "Chronological view". Below "Info" is a "Results" section showing "4 methods, 4 passed" and a list of passed methods: "Menuclick", "OpenHomepage", "cookieconsent", and "searchproduct". The right section displays the test results for the suite "assessment.main.Home". It lists the methods "Menuclick", "OpenHomepage", and "cookieconsent" under the category "assessment.main.Home". Below this, it shows the category "assessment.main.Category" with the method "searchproduct".