Parameterized Test with @CsvFileSource

This lesson demonstrates the use of @CsvFileSource to pass different arguments to @ParameterizedTest.

WE'LL COVER THE FOLLOWING

@CsvFileSource

`@CsvFileSource` allows you to use CSV files from the classpath. This csv file gets picked up from the classpath at the time of running test case and each line from a `CSV` file results in one invocation of the parameterized test. We can also provide a `number of lines` to skip from top to take comma-separated values.

Let's look at a demo.

Step 1 - Let's assume that we have to write a parameterized test that takes values from `@CsvFileSource`.

Step 2 - We create a csv file by name, `capitals.csv`. It has comma-separated values of countries and their capitals.

Step 3 - We will keep this csv file on the classpath.

capitals.csv

```
Country, Capital
India, New Delhi
France, Paris
United States, Washington D.C.
United Kingdom, London
```

Step 4 - We create a test class by name, `CsvFileSourceTest.java`.

Step 5 - It contains a test method by name, `testWithCsvFileSource` method. In

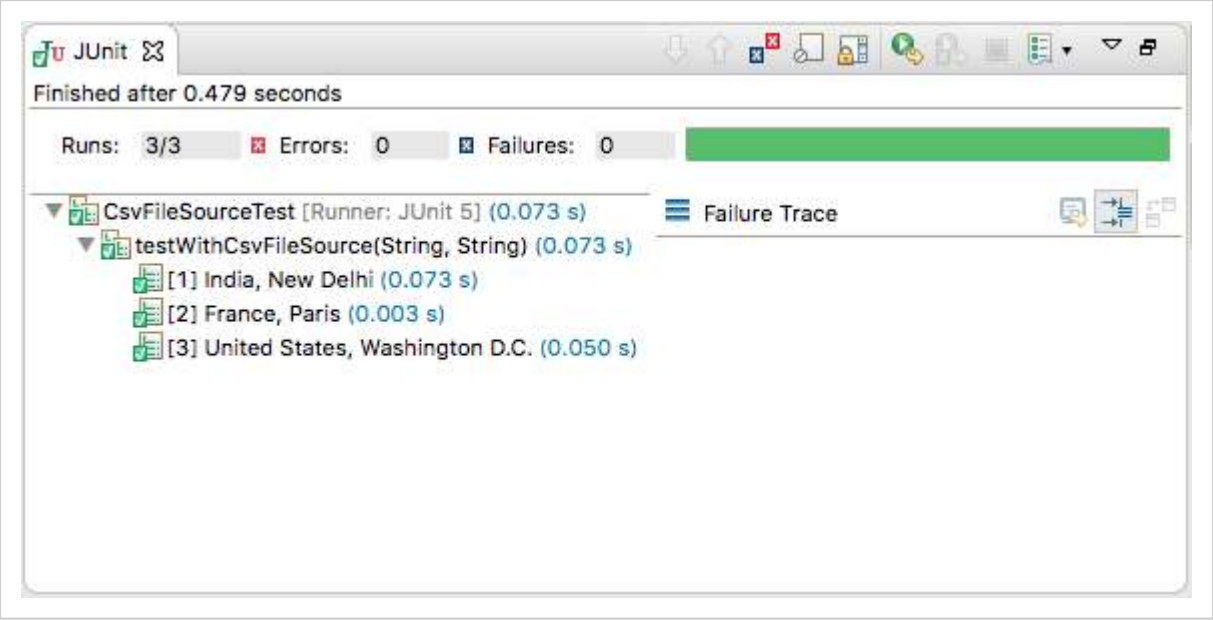
order to provide different parameters/values to the same test method, this method is marked as `@ParameterizedTest` instead of `@Test`.

Step 6 - In order to provide different and multiple values through csv file source we mark this test method with `@CsvFileSource` annotation. This annotation takes `resources` which is the path to the csv file and `numLinesToSkip` which an integer value, to let test method skip those many lines while providing arguments to `@ParameterizedTest`.

Let's see the test class below.

CsvFileSourceTest.java

```
package com.hubberspot.junit5.parameterized;

import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvFileSource;

class CsvFileSourceTest {

        @ParameterizedTest
        @CsvFileSource(resources = "/capitals.csv", numLinesToSkip = 1)
        void testWithCsvFileSource(String country, String capital) {
            assertNotNull(country);
            assertNotNull(capital);
        }

}
```



Output of @ParameterizedTest demo

Above image demonstrates the working of `@ParameterizedTest`. As we have provided 4 different csv file source values which are comma-separated, so the

provided 4 different csv file source values which are comma separated, so the first argument to test method is a String which is country and second

argument is a String which is capital, therefore the test case ran 4 times. Also, all string values provided by csv file source are not null, therefore `assertNotNull` passes for all values passed.

In the next lesson we will be studying `Assumptions` in Junit 5.