Tags Logic in Cucumber

@tag1 Runs all scenarios with @tag1

@tag1 or @tag2 Runs scenarios with @tag1 or @tag2

@tag1 and @tag2 Runs scenarios that have both @tag1 and @tag2

not @tag1 Runs all scenarios except those with @tag1

tags = "(@T1 or @T2 Or @T3) and (not @ignore)",

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

@CucumberOptions(

features = "src/test/resources/features",

glue = "stepDefinitions",

plugin = {"pretty", "html:target/cucumber-reports"},

tags = "@SmokeTest",

monochrome = true

)

public class TestRunner extends AbstractTestNGCucumberTests {

}

When monochrome = true:

Removes all ANSI color codes from the console output (i.e., no colored text), making it plain and more readable/log-friendly.

When monochrome = false (default):

Output may include special characters or color codes, which might look garbled in some terminals or logs.

Always recommended to set monochrome = true

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

@ignore ========

to temporarily disable scenarios you don’t want to run — like ones that are incomplete, flaky, or under development

@s1

Scenario: Valid login

@T1

Given user is on login page

When user enters valid credentials

Then user should be logged in successfully

@s2

Scenario: Invalid login

@T2 @ignore

Given user is on login page

When user enters wrong password

Then user should see an error

Hooks:

https://cucumber.io/docs/cucumber/api/#hooks

package Com.Prog.Cucumber.Hooks;

import io.cucumber.java.After;

import io.cucumber.java.AfterAll;

import io.cucumber.java.AfterStep;

import io.cucumber.java.Before;

import io.cucumber.java.BeforeAll;

import io.cucumber.java.BeforeStep;

import io.cucumber.java.Scenario;

public class Hooks

{

@Before

public void setUp()

{

System.out.println(" Befroe exected...");

}

@After

public void tearDown()

{

System.out.println(" @After exected...");

}

@BeforeAll

public static void before\_or\_after\_all1()

{

System.out.println(" @@BeforeAll exected...");

}

@AfterAll

public static void before\_or\_after\_all2()

{

System.out.println(" @@@AfterAll exected...");

}

// @BeforeStep

// public void doSomethingBeforeStep(Scenario @s2){

// }

//

// @AfterStep

// public void doSomethingAfterStep(Scenario scenario){

// }

}

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

dryRun: check whether all feature file steps have corresponding step definitions

Note : not extecute the stepdefination code neither the Hooks.

//@CucumberOptions(

// dryRun=false,

// features= {".\\src\\test\\java\\Com\\Proj\\Cucumber\\FeaturePkg\\login.feature",

// ".\\src\\test\\java\\Com\\Proj\\Cucumber\\FeaturePkg\\login2.feature"},

// glue={"Com\\Proj\\Cucumber\\definations","Com\\Prog\\Cucumber\\Hooks"},

// plugin = {"pretty", "html:target/cucumber-report.html"}

//

//

// )

@CucumberOptions(

dryRun=true,

features= {".\\src\\test\\java\\Com\\Proj\\Cucumber\\FeaturePkg\\login.feature"},

glue={"Com\\Proj\\Cucumber\\definations","Com\\Prog\\Cucumber\\Hooks"},

plugin = {"pretty", "html:target/cucumber-report.html"}

)

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

**Parallel Execution using Testng & Junit in BDD:**

**JUnit 5:**

**Cucumber Scenarios can be executed in parallel**

**Junit4:**

JUnit the **feature files are run in parallel rather than scenarios**, which means **all the scenarios in a feature file will be executed by the same thread**

TestNg:

**Cucumber Scenarios can be executed in parallel**

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

Feature: Store login

This feature deals with the login functionality.

Scenario: Login with correct username and correct password

Given I navigate to login page

#And I entered username and password

When I entered "admin1" and "pass1"

And I clicked signin button

Then Its navigated to the home page

Scenario: Login with Incorrect username and correct password

Given I navigate to login page

#And I entered username and password

When I entered "&\*^\*&" and "pass1"

And I clicked signin button

Then Its navigated to the home page

Scenario: Login with correct username and Incorrect password

Given I navigate to login page

#And I entered username and password

When I entered "admin1" and "^&&\*&\*"

And I clicked signin button

Then Its navigated to the home page

Scenario / Testcases Parallel execution:

1. using @DataProvider...., testNg.xml, throught POM.xml

@CucumberOptions(

dryRun=false,

features= {".\\src\\test\\java\\Com\\Proj\\Cucumber\\ParallelRunDemo"},

glue={"Com\\Proj\\Cucumber\\ParallelRunDemo"},

plugin = {"pretty", "html:target/cucumber-report.html"

}

)

public class Runner extends AbstractTestNGCucumberTests

{

@DataProvider(parallel=true) // parallel run

public Object[][] scenarios()

{

return super.scenarios();

}

}

---------

public class Runner extends AbstractTestNGCucumberTests

{

@DataProvider(parallel=false) // sequential run

public Object[][] scenarios()

{

return super.scenarios();

}

}

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<suite name="Suite" data-provider-thread-count="2">

<test thread-count="5" name="Test" >

<classes>

<class name="Com.Proj.Cucumber.ParallelRunDemo.SampleRunner"></class>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

---------

PoM.xml:

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.3.0</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>C:\Users\Achala\eclipse-workspace\Loans\src\test\java\Com\Proj\Cucumber\ParallelRunDemo\Sample.xml</suiteXmlFile>

</suiteXmlFiles>

</configuration>

</plugin>

-----

pom.xml ==> runAs ==> install

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

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

Parallel execution using Junit:

package Com.Proj.Cucumber.ParallelRunDemo;

import org.junit.runner.RunWith;

import io.cucumber.junit.Cucumber;

import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)

@CucumberOptions(

dryRun=false,

features= {".\\src\\test\\java\\Com\\Proj\\Cucumber\\ParallelRunDemo"},

glue={"Com\\Proj\\Cucumber\\ParallelRunDemo"},

plugin = {"pretty", "html:target/cucumber-report.html",

"timeline:target/cucumber"

}

)

public class SampleJunitRunner

{

}

---------------

pom.xml:

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.3.0</version>

<configuration>

<includes>\*\*\SampleJunitRunner.java</includes>

<parallel>methods</parallel>

<threadCount>2</threadCount>

</configuration>

</plugin>

----------

pom.xml ==> runAs ==> install