



30 ARALIK 2020
DERS 29

Hooks
Paralel Testing

TECHPROED

Mehmet BULUTLUOZ
Elektronik Muh.



Yeni Raporlar Ekleme

- Plugin ekleyerek yeni raporlar da olusturabiliriz
- Tester'lar icin onemli olan rapor Html olsa da json ve xml formatinda da rapor almak mumkundur.
- Ayrica maven-cucumber-reporting plugin yuklemek istersek pom.xml'e plugin ekleyebiliriz

```
@CucumberOptions(  
    plugin={"html:target\\cucumber-reports.html",  
          "json:target/json-reports/cucumber.json",  
          "junit:target/xml-report/cucumber.xml" },  
    features = "src/test/resources/features",  
    glue = "stepdefinitions",  
    dryRun = false,  
    tags="@amazon"  
)
```

```
@RunWith(Cucumber.class)  
@CucumberOptions(  
    plugin={"html:target/cucumber_rapor.html"},  
    features = "src/test/resources/features",  
    glue="stepdefinitions",  
    tags="@amazon",  
  
    dryRun = false  
)
```




Hooks ve Screen Shot Ekleme

- Cucumber hooks, senaryolardan önce veya sonra çalışan kod bloklarına sahip olan bir classtır. (Daha önce kullandığımız TestBase gibi)
- @Before ve @After annotation'ları kullanılarak kodları projemizde ve step definitionlarda kullanabiliriz.
- Cucumber hooks, kod çalışma akışını daha iyi yönetmemizi kolaylaştırır ve kod fazlalığını azaltmamıza yardımcı olur.



@After

```
public void tearDown(Scenario scenario){  
    final byte[] screenshot=((TakesScreenshot) Driver.getDriver()).getScreenshotAs(OutputType.BYTES);  
    if (scenario.isFailed()) {  
        scenario.attach(screenshot, "image/png", "screenshots");  
    }  
    Driver.closeDriver();  
}
```



cucumber

Paralel Testing

- Paralel testing: Birden fazla browser'in es zamanli calistirilmasidir.
- Cucumber ile parallel test calistirmek testing'ye gore daha zordur.
- Paralel calistirabilmek icin birden fazla Runner Class'ina ihtiyacimiz var
- Cucumber'da, testleri paralel olarak calistirabilmek icin bazı eklentilere(plugin) ve yapilandirmalara ihtiyacimiz var.
- Pom da yaptigimiz ayarlamalardan sonra testleri Runner'dan degil Terminalden **"mvn clean verify"** kodunu yazarak calistiracagiz

TECHPROED



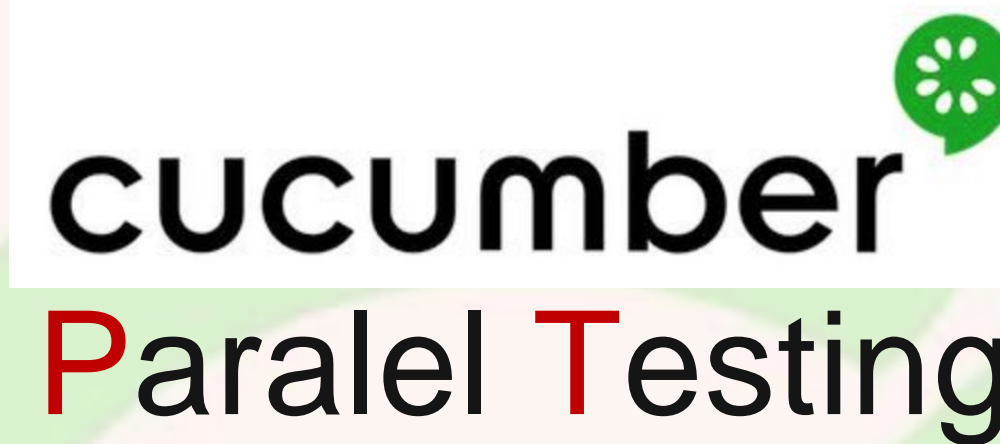
Paralel Testing

1. Birden fazla runner classı ekleyin. Aynı anda çalıştırmak istediğiniz kadar Runner Class'ına sahip olmalısınız. Class isimleri TestRunner ile bitmelidir

- a. SmokeTestRunner
- b. FirstTestRunner
- c. SecondTestRunner

2. maven-failsafe-plugin eklentisi ekleyin. (Belirli testler başarısız olduktan sonra testleri çalıştırmaya devam için.)

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-failsafe-plugin</artifactId>
  <version>3.0.0-M1</version>
  <configuration>
    <testFailureIgnore>true</testFailureIgnore>
    <skipTests>>false</skipTests>
    <includes>
      <include>**/runners/*TestRunner*.java</include>
    </includes>
  </configuration>
</plugin>
```



3. maven-surefire-plugin ekleyin ve yapılandırın. Paralel test için gerekli eklentidir

<plugin>

<groupId>org.apache.maven.plugins**</groupId>**

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

<version>3.0.0-M1**</version>**

<configuration>

<parallel>classes**</parallel>**

<forkMode>perthread**</forkMode>**

<threadCount>4**</threadCount>**

<reuseForks>false**</reuseForks>**

<argLine>-Duser.language=en**</argLine>**

<argLine>-Xmx1024m**</argLine>**

<argLine>-XX:MaxPermSize=256m**</argLine>**

<argLine>-Dfile.encoding=UTF-8**</argLine>**

<useFile>false**</useFile>**

<includes>

<include>**/*.java**</include>**

</includes>

<testFailureIgnore>true**</testFailureIgnore>**

</configuration>

</plugin>



Paralel Testing

4. maven-cucumber-reporting plugin ekle. Gelismis rapor icin gereklidir

```
<plugin>
  <groupId>net.masterthought</groupId>
  <artifactId>maven-cucumber-reporting</artifactId>
  <version>5.0.0</version>
  <executions>
    <execution>
      <id>execution</id>
      <phase>verify</phase>
      <goals>
        <goal>generate</goal>
      </goals>
      <configuration>
        <projectName>cucumber-jvm-example</projectName>
        <outputDirectory>${project.build.directory}</outputDirectory>
        <inputDirectory>${project.build.directory}</inputDirectory>
        <jsonFiles>
          <param>**/json-reports/*.json</param>
        </jsonFiles><classificationFiles>->
        <param>sample.properties</param>
        <param>other.properties</param>
      </classificationFiles>
    </configuration>
  </execution>
</executions>
</plugin>
```

cucumber

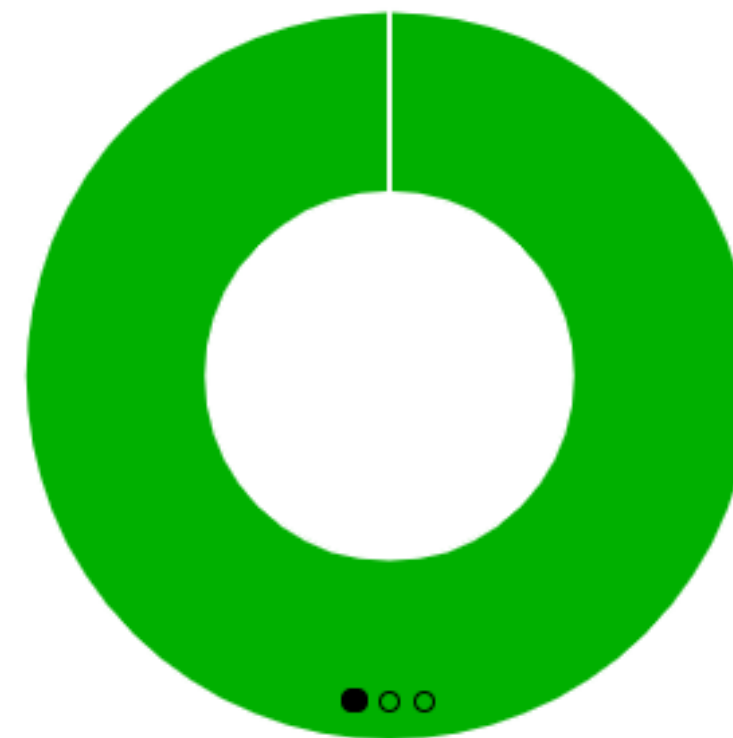
Parallel Testing

Project	Number	Date
cucumber-jvm-example	1	01 Jan 2021, 15:21

Features Statistics

The following graphs show passing and failing statistics for features

Features



Feature	Steps						Scenarios			Features	
	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
US1001 amazon page search	10	0	0	0	0	10	1	0	1	30.902	Passed
US1002_amazon_search_background	12	0	0	0	0	12	3	0	3	41.928	Passed
US1004_amazon_search_scenario_outline	16	0	0	0	0	16	4	0	4	53.219	Passed
	7	0	0	0	0	7	1	0	1	28.033	Passed
US1009 Ck Hotels Log	7	0	0	0	0	7	1	0	1	15.471	Passed
	52	0	0	0	0	52	10	0	10	2:49.553	5
	100.00%	0.00%	0.00%	0.00%	0.00%		100.00%	0.00%			100.00%

cucumber

Parallel Testing

Project	Number	Date
cucumber-jvm-example	1	01 Jan 2021, 15:21

Tags Statistics

The following graph shows passing and failing statistics for tags



Tag	Steps						Scenarios			Features	
	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
@amazon	35	0	0	0	0	35	8	0	8	1:35.482	Passed
@amazonarama	3	0	0	0	0	3	1	0	1	3.187	Passed
@mehmet	3	0	0	0	0	3	1	0	1	3.865	Passed
@smoke	14	0	0	0	0	14	2	0	2	43.504	Passed
@smoketest	6	0	0	0	0	6	2	0	2	7.052	Passed
	61	0	0	0	0	61	14	0	14	2:33.090	5
	100.00%	0.00%	0.00%	0.00%	0.00%		100.00%	0.00%			100.00%

cucumber

Parallel Testing

Project	Number	Date
cucumber-jvm-example	1	01 Jan 2021, 15:21

Steps Statistics

The following graph shows step statistics for this build. Below list is based on results. step does not provide information about result then is not listed below. Additionally @Before and @After are not counted because they are part of the scenarios, not steps.

Implementation	Occurrences	Average duration	Max duration	Total durations	Ratio
stepdefinitions.AmazonStepDefinitions.flower_icin_arama_yapar()	2	3.892	4.117	7.784	100.00%
stepdefinitions.AmazonStepDefinitions.icinAramaYapar(java.lang.String)	4	3.460	4.144	13.842	100.00%
stepdefinitions.AmazonStepDefinitions.iphone_icin_arama_yapar()	2	3.216	3.448	6.432	100.00%
stepdefinitions.AmazonStepDefinitions.kullaniciSayfayikapatir()	10	0.088	0.119	0.884	100.00%
stepdefinitions.AmazonStepDefinitions.kullanici_amazon_anasayfaya_gider()	10	8.741	13.112	1:27.416	100.00%
stepdefinitions.AmazonStepDefinitions.sonucunIcerdiginiTestEder(java.lang.String)	4	0.118	0.177	0.474	100.00%
stepdefinitions.AmazonStepDefinitions.sonucun_flower_icerdigini_test_eder()	2	0.165	0.206	0.331	100.00%
stepdefinitions.AmazonStepDefinitions.sonucun_iphone_icerdigini_test_eder()	2	0.122	0.142	0.244	100.00%
stepdefinitions.AmazonStepDefinitions.sonucun_tea_pot_icerdigini_test_eder()	2	0.112	0.152	0.225	100.00%
stepdefinitions.AmazonStepDefinitions.tea_pot_icin_arama_yapar()	2	4.324	4.504	8.649	100.00%
stepdefinitions.BestbuyStepDefinitions.kullanici_anasayfaya_gider(java.lang.String)	2	13.412	14.311	26.825	100.00%
stepdefinitions.CkHotelsStepDefinitions.gecerli_password_girer()	1	0.188	0.188	0.188	100.00%
stepdefinitions.CkHotelsStepDefinitions.gecerli_username_girer()	1	0.230	0.230	0.230	100.00%
stepdefinitions.CkHotelsStepDefinitions.gecersizPasswordGirer()	1	0.190	0.190	0.190	100.00%
stepdefinitions.CkHotelsStepDefinitions.gecersizUsernameGirer()	1	0.221	0.221	0.221	100.00%
stepdefinitions.CkHotelsStepDefinitions.log_in_yazisina_tiklar()	2	1.145	1.200	2.291	100.00%
stepdefinitions.CkHotelsStepDefinitions.login_butonuna_basar()	2	6.593	12.063	13.186	100.00%
stepdefinitions.CkHotelsStepDefinitions.sayfayaGirisYapilamadiginiKontrolEder()	1	0.104	0.104	0.104	100.00%
stepdefinitions.CkHotelsStepDefinitions.sayfaya_giris_yaptigini_kontrol_eder()	1	0.037	0.037	0.037	100.00%
19	52	3.260	1:27.416	2:49.553	Totals