|  |  |
| --- | --- |
| **CUCOMBER Framework oluşturma.** | **Proje Açma Aşamaları** ve Temel Kodları:  Behavior-Driven Development (BDD) |
| 1. **File**-new-Project-select Maven-Next-name”ornekCucomber”-finish |  |
| 2. **pom.xml:** selenium-java(4,1,3)/ webdrivermanager(5.1.0)/ cucumber-java(7.2.3)/ cucumber-junit(7.2.3) eklenir. | **Raporlama için:** </dependencies> <build> <plugins>maven-cucumber-reporting(5.0.0)/ |
| 2a**. pom.xml içine paralel test için**;  Browser sayısı kadar TestRunner clası ve <threadCount>browser sayısı kadar</threadCount> olması kontrol edilir.  Yani TestRunner clası sayısı ile threadCount sayısı birbirine eşit olmalıdır. | **Paralel test için:** maven-failsafe-plugin(3.0.0-M1)/ maven-surefire-plugin(3.0.0-M1) /  **Paralel test için:** maven-**failsafe**-plugin(3.0.0-M1)/  <includes>  <include>\*\***/runners/\*TestRunner\*.**java</include> </includes>  maven-**surefire**-plugin(3.0.0-M1) /  <threadCount>**2**</threadCount> |
| 3. **resources**: src-test-new-directory-name”resources” |  |
| 3a. **features**: test-resources-new- directory-name” features” / |  |
| 3b. **feature**: new-file-newfile “name.feature”  **Yeşil çıkmazsa:** File-setting-plugin-cucumberjava-install | Feature: US1009 DataTables sitesine giris yapar  @hmc  Scenario: TC14 kayit girisi   When kullanici "datatableUrl" icin arama yapar  Then new butonuna basar |
| 4. **configurations.properties**: Proje adı üstünde sağ tıkla-new-file-name” configurations.properties” | browser= "chrome'  dataTablesUrl=https://editor.datatables.net/ |
| 5. **Packagelar**: src-test-java altında utilities-stepDefinations-runners-pages olusturulur |  |
| 5a1. **Utilities/ConfigReader**: new- java class-“ConfigReader” | public class ConfigReader {  public static Properties *properties*; static{String dosyaYolu="configuration.properties";  try{FileInputStream fis=new FileInputStream(dosyaYolu); *properties*=new Properties();*properties*.load(fis); } catch (IOException e) {e.printStackTrace();}}  public static String getProperty(String key){  return *properties*.getProperty(key); } } |
| 5a2. **Utilities/Driver**: new-java class-Driver | public class Driver {  private Driver(){ }  static WebDriver *driver*;  public static WebDriver getDriver(){ if (*driver*==null) { WebDriverManager.*chromedriver*().setup(); *driver* = new ChromeDriver();  *driver*.manage().window().maximize(); *driver*.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(15)); } return *driver*;}  public static void closeDriver(){  if (*driver*!=null){ *// driver'a deger atanmissa  driver*.close(); *driver*=null; } } |
| 5b. **Runners**: new-java class-“Runner” | @RunWith(Cucumber.class) @CucumberOptions(  features = “src/test/resources/features”,  glue = “stepDefinitions”,  tags = “@hmc”,  dryRun = false) public class Runner { } |
| 5b1. **Raporlama için** “Runner” clasının adı “TestRunner” olarak değiştirilir.  Ve plugin kodları eklenir. | @RunWith(Cucumber.class) @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”,  tags = “@hmc”,  dryRun = false) public class TestRunner { } |
| 5b2. **Paralel test için:** “Runner” clasının adı “TestRunner” ve “TestRunner2” olmak üzere 2 (browser sayisi kadar) adet olarak TestRunner classı oluşturulur.  Raporlarında üst üste yazılmaması için de “TestRunner2” Plugin’lere “2” yazılır. Tags=@hmc2’ da 2 eklenir. | @RunWith(Cucumber.class) @CucumberOptions(  **plugin=**{"html:target/cucumber-reports**2**.html",  "json:target/json-reports**2**/cucumber.json",  "junit:target/xml-report**2**/cucumber.xml" },  features = “src/test/resources/features”,  glue = “stepDefinitions”,  tags = “@hmc2”,  dryRun = false) public class TestRunner2 { } |
| 5b3. Pom.xml içine paralel test için;  Browser sayısı kadar TestRunner clası olması kontrol edilir.  Yani TestRunner clası sayısı ile threadCount sayısı birbirine eşit olmalıdır. | **Paralel test için:** maven-failsafe-plugin(3.0.0-M1)/  <includes>  <include>\*\*/runners/\*TestRunner\*.java</include> </includes> |
| 5c. **Pages**:-new-java class-name ”HerokuappPages” | public class DataTablesPage {  public DataTablesPage(){ PageFactory.*initElements*(Driver.*getDriver*(),this);}  @FindBy(xpath = “//button[@class=’dt-button buttons-create’]”) public WebElement newBotonu; @FindBy(xpath = “//input[@id=’DTE\_Field\_first\_name’]”) public WebElement firsNameKutusu; |
| 5d. **stepDefinations**: new-java class-“ DataTableStepDefination” | public class DataTableDefinations { DataBaseTablePages dataBaseTablePages=new DataBaseTablePages();  @Then("kullanici Nutella icin arama yapar") public void kullanici\_nutella\_icin\_arama\_yapar()  {amazonPage.aramaKutusu.sendKeys("Nutella"+ Keys.*ENTER*);} |
| 5d1. Raporlama için: “**Hooks**” class’ı açılır.  stepDefinations-new-javaclass-name”Hooks” | public class Hooks {  @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*();}} |
| **6 Paralel testin çalıştırılması** | **Terminal-mvn clean verify-Enter** |
|  |  |

**Pom.xml**

*<?*xml version="1.0" encoding="UTF-8"*?>*<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>org.example</groupId>  
 <artifactId>com.batch59Cucumber</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>17</maven.compiler.source>  
 <maven.compiler.target>17</maven.compiler.target>  
 </properties>  
<dependencies>  
 *<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->* <dependency>  
 <groupId>org.seleniumhq.selenium</groupId>  
 <artifactId>selenium-java</artifactId>  
 <version>4.1.3</version>  
 </dependency>  
  
 *<!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->* <dependency>  
 <groupId>io.github.bonigarcia</groupId>  
 <artifactId>webdrivermanager</artifactId>  
 <version>5.1.0</version>  
 </dependency>  
 *<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->* <dependency>  
 <groupId>io.cucumber</groupId>  
 <artifactId>cucumber-java</artifactId>  
 <version>7.2.3</version>  
 </dependency>  
 *<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-junit -->* <dependency>  
 <groupId>io.cucumber</groupId>  
 <artifactId>cucumber-junit</artifactId>  
 <version>7.2.3</version>  
 <scope>test</scope>  
 </dependency>  
  
  
</dependencies>  
 <build>  
 <plugins>  
 <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>  
 <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>  
 <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>2</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>\*\*/runners/\*TestRunner\*.java</include>  
 </includes>  
 <testFailureIgnore>true</testFailureIgnore>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>