
TestNG

23 ARALIK 2020
DERS 24

Kahoot

TestNG Ornek Proje Olusturma

Mehmet BULUTLUOZ
Elektronik Muh.

TestNG Örnek Proje Oluşturma

1. File – New – Project e tikliyoruz
2. Maven'i seciyoruz
3. Name'e projemizin ismini yazıyoruz
4. Cikan Alert mesajinda New Window veya This Window secilebilir
5. Pom xml'imizi düzenliyoruz

a) `<properties>`
`<maven.compiler.source>1.8</maven.compiler.source>`
`<maven.compiler.target>1.8</maven.compiler.target>`
`</properties>`
*** bu kod Javanin sürümüyle alakali sorunlari halletmeye yariyor

b) `<dependencies>`
Kutuphanelerimizi bu tag'lar arasina yaziyoruz
`</dependencies>`

TestNG Ornek Proje Oluşturma

6) <https://mvnrepository.com/> a gidip kutuphanelerimizi tek tek alıyoruz

a) Selenium-Java Kutuphanesi

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>3.141.59</version>
</dependency>
```

b) WebDriverManager Kutuphanesi

```
<!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
<dependency>
  <groupId>io.github.bonigarcia</groupId>
  <artifactId>webdrivermanager</artifactId>
  <version>4.2.0</version>
</dependency>
```

TestNG Ornek Proje Olusturma

c)Testng Kutuphanesi

```
<!-- https://mvnrepository.com/artifact/org.testng/testng -->  
<dependency>  
  <groupId>org.testng</groupId>  
  <artifactId>testng</artifactId>  
  <version>7.1.0</version>  
  <scope>test</scope>  
</dependency>
```

d)Java Faker Kutuphanesi

```
<!-- https://mvnrepository.com/artifact/com.github.javafaker/javafaker -->  
<dependency>  
  <groupId>com.github.javafaker</groupId>  
  <artifactId>javafaker</artifactId>  
  <version>1.0.2</version>  
</dependency>
```

TestNG Ornek Proje Olusturma

e)Apachi-POI Kutuphanesi(2 tane)

```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi</artifactId>
  <version>4.1.2</version>
</dependency>
```

```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi-ooxml</artifactId>
  <version>4.1.2</version>
</dependency>
```

TestNG Ornek Proje Olusturma

- 7) Pom Dosyasini oluşturma işlemimiz bitti. Kutuphaneler kirmizi renkte olabilir. Sag tarafta Maven yazan sekmede Reload oklarına tiklayip beklediğimiz de hata gitmiş oluyor
- 8) Kullanıcının gordugu arayuzde test ederiz. (UI)
Kullanacagimiz paketleri uygun isimlerde test-Java bolumun içerisinde oluşturunuz
- 9) Java ya sag tiklariz new package – com (paketin ismi) yazariz
- 10) com package'ina sag tiklariz – new package – techproed (paketin ismi) yazariz
- 11) artik projemiz com. techproed

Bu package'in altina frameworkumuzun package'larini yolusturuyoruz. Bunlar

A-pages

B-smokeTest

C-tests

D-utilities

TestNG Ornek Proje Olusturma

12) Resources paketi olusturma:

Java'ya sag tikliyoruz-new-package -->resources (yeni package)

Bu resources paketinin altina dokumanlarimizi copy-paste ederiz

13) configuration.properties dosyasi olusturma

En yukarda Projemize sag tikliyoruz new- File ' a tikliyoruz

Ismi önemli değil ama uzantisi MUTLAKA .properties olmalı

İsmi configuration.properties yazıyoruz

Bu dosyanin içine Data'larimizi key=value seklinde yazıyoruz

Ck_hotels_url=<http://qa-environment.crystalkeyhotels.com>

Ck_hotels_user=managerCk_hotels_password=Manager1!

14) ConfigReader Class'ı olusturma

utilities package inin altinda ConfigReader Classi oluşturunuz. Bu class

configuration.properties deki dosyalarimizi okumak için bir araci

TestNG Ornek Proje Olusturma

15) ConfigReader Classinda :

1-ilk yapacagimiz sey Instance olarak Properties objesi olusturmak. Bu objeyi static blok icinde kullanacagimdan static yapmam gerek

Bu objeyi sadece bu class ta kullanacagim icin private yapmamiz önerilir

2-Properties objesini kullanmak uzere bir static blok kurmaliz. neden static? Cunku her zaman ilk static block calisir

```
public class ConfigReader {  
  
    private static Properties properties;  
  
    static {  
        String path="configuration.properties";  
        try {  
            FileInputStream fileInputStream=new FileInputStream(path);  
            properties=new Properties();  
            properties.load(fileInputStream);  
  
            fileInputStream.close();  
        } catch (FileNotFoundException e) {  
            e.printStackTrace();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
  
    public static String getProperty(String key){  
  
        return properties.getProperty(key);  
    }  
}
```


TestNG Ornek Proje Olusturma

16) Driver class'ini düzenliyoruz

Singleton class : object olusturulmasi kontrol altına alınan (genelde izin verilmeyen) classdir. Bunun için baska classlarda Driver clasindan obje uretmemizi saglayan default constructor'i görünür sekilde yazip access modifier'i private yapariz

Bu class'da test class'larimizda kullanacagimiz driver'i olusturacak ve kapatacak `getDriver()` ve `closeDriver()` methodlarini olusturuyoruz

Bu methodlari static yaparak obje olusturmadan Class adi ile cagirmak için kullanisli hale getiriyoruz

```
public class Driver {  
    private Driver(){  
    }  
    static private WebDriver driver;  
    static public WebDriver getDriver(){  
        if(driver==null){  
            switch (ConfigReader.getProperty("browser")){  
                case "chrome":  
                    WebDriverManager.chromedriver().setup();  
                    driver=new ChromeDriver();  
                    break;  
                case "firefox":  
                    WebDriverManager.firefoxdriver().setup();  
                    driver=new FirefoxDriver();  
                    break;  
                case "safari":  
                    WebDriverManager.getInstance(SafariDriver.class);  
                    driver=new SafariDriver();  
                    break;  
                case "opera":  
                    OperaDriverManager.operadriver().setup();  
                    driver=new OperaDriver();  
                    break;  
            }  
        }  
        driver.manage().window().maximize();  
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  
        return driver;  
    }  
  
    static public void closeDriver(){  
        if (driver != null){  
            driver.close();  
            driver=null;  
        }  
    }  
}
```

TestNG Örnek Proje Oluşturma

18) pages package inin altında kullanacağımız her websayfasi için bir page Class'i oluştururuz

a) Bu class'da ilk yapmamız gereken test class'larında bu class'dan obje üretebilmemiz için gerekli olan Constructor'i oluşturmaktır.

```
public CkHotelsHomePage(){  
    PageFactory.initElements(Driver.getDriver(), page: this);  
}
```

b) Ardında Locate işlemlerimizin tamamını yazıyoruz bu sayfaya

```
@FindBy(linkText="Log in")  
public WebElement ilkLogIn;  
  
@FindBy(id="UserName")  
public WebElement userNameTextBox;  
  
@FindBy(id="Password")  
public WebElement passwordTextBox;  
  
@FindBy(id="btnSubmit")  
public WebElement loginButonu;
```