



2 ARALIK 2020  
DERS 9

TestNG Proje Olusturma  
Handle Dropdown

Mehmet BULUTLUOZ  
Elektronik Muh.

TECHPROED

---

# TestNG

## Önceki Dersten Aklımızda Kalanlar

- 1- Assertion : Testin istedigimiz sonucu verip vermedigini control eden, ve bazı durumlarda sonucu rapor eden static methodlardir. Ve bundan önce if else ile yaptigimiz dogrulama ve raporlama islemlerinin gelismis halidir.
- 2- Assert.assertEquals(Beklenen sonuc , gerceklesen sonuc) Yazilim testinin temel amaci olan, beklenen sonuc ile gerceklesen sonucun esit olup olmadigini kontrol eder ve eger esit degilse aldiklari degerleri rapor eder
- 3- Assert.assertTrue(boolean condition) yazilan kosulun gerceklesmesi durumunda Testi PASS yapar. Dolayisiyla positive testlere daha uygundur.
- 4-Assert.assertFalse(boolean condition) yazilan kosulun gerceklesmemesi durumunda testi PASS yapar. Dolayisiyla negative testlere daha uygundur.

TECHPROED

# TestNG

- **Junit**'in yeni versiyonudur.
- İsmi **Next Generation Test** kelimelerinden türetilmiş , Cédric Beust tarafından oluşturulmuştur.
- Açık Kaynak kodludur.
- TestNG bir test kütüphanesidir.
- TestNG sadece JAVA ile çalışır ve JDK 7 ve daha üst versiyonları gereklidir
- TestNG ile ilgili dokümanlara aşağıdaki adresten ulaşılabilir

<https://testng.org/doc/documentation-main.html>

TECHPROED

# TestNG

- **TestNG** tester'lara daha fazla control imkani verir ve testleri daha etkili yapmamizi saglar.
- Tester'lar **TestNG**'yi etkili bir framework tasarlamak ve test case'leri TestNG annotation'ları ile organize etmek için kullanırlar.
- TestNG ile daha fazla before ve after methodlari kullanilabilir
- Test caseleri siralama ozelligi (**priority**) ve test caselerin birbirine bagimlilik (b)**dependsOnMethod**) bize testleri organize etmekte yardım eder.
- Yazılan test case'lerin paralel olarak çalıştırılmasına,birden fazla kullanilmasına imkan tanir.
- Error mesajlarının daha detaylı bir şekilde gösterilmesini sağlar.
- Multi-Browser Test
- Kullanisli HTML veya xml raporlari olusturmaya yarar

# TestNG

## Nasil Yuklenir?

- eclipse veya intellij için TestNG eklemenin farklı yolları vardır. Maven kullandığımız için dependency biçimi olarak ekleyeceğiz.
- <https://mvnrepository.com/> , adresine gidin
- TestNG'yi aratın ve dependency'i pom.xml'e ekleyin.

```
<dependency>
```

```
    <groupId>org.testng</groupId>
```

```
<d>
```

```
    <artifactId>testng</artifactId>
```

```
>
```

```
    <version>7.0.0</version>
```

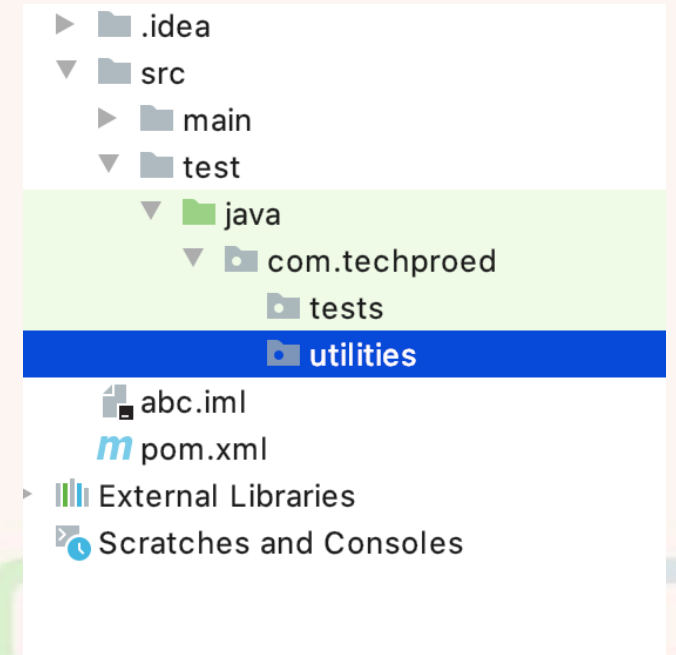
```
    <scope>test</scope>
```

```
</dependency>
```

# TestNG

## Proje Olusturma

1. Create Project: File -> New -> Project-> Select maven -> click next
2. Isim:**batch11testNGframework**->finish->cikarsa **EnableAutoImport**'u tiklayin
3. java'ya Right click yapin ve "create package" secin.  
package ismi : **com.techproed**
4. **com.techproed**'a Right click yapin ve "create package" secin.  
package ismi : **com.techproed.tests**
5. Bir daha **com.techproed**'a Right click yapin ve "create package" secin.  
package name : **com.techproed.utilities**
6. Right click on tests package and create a java class.  
Class name: **Priority**
7. Add the dependencies on you pom.xml file:  
Get the dependencies mvnrepository.com
  - a. selenium java,
  - b. webdrivermanager,
  - c. testNG



# TestNG

## @Test annotation

- En çok kullanılan TestNG notasyonudur.
- Method'u test case olarak işaretler ve çalıştırır. Ayrıca bir Main Method'a ihtiyaç duymaz.

### @Test

```
public void test1(){
```

```
WebDriverManager.chromedriver().setup();
```

```
WebDriver driver= new
```

```
ChromeDriver();
```

```
driver.get("https://www.google.com/");
```

```
}
```

```
import org.testng.annotations.*;

public class TestNGFirst {
    @BeforeMethod @BeforeClass @BeforeTest @BeforeSuite//use only one
    public void setUp(){
        System.out.println("Contains what ever you need before executing a test" +
            "//such as driver set up, waits, maximize, and, go to main URL etc.");
    }

    @Test
    public void test1() { System.out.println("Test case 1"); }

    @Test
    public void test2() { System.out.println("Test case 2"); }

    @AfterMethod @AfterClass @AfterTest @AfterSuite//use only one |
    public void tearDown(){ System.out.println("Contains what ever you do at the end of the test case" +
        "such as closing/quitting driver, getting screenshots, generating reports, etc."); }
}
```



# TestNG

## @Before @After Annotations

- @BeforeSuite @AfterSuite => runs before/ after all tests in this suite
- @BeforeTest @AfterTest. => run before/after all the test methods after Test
- @BeforeGroups @AfterGroups =>run before/after any specific test group
- @BeforeClass @AfterClass =>run before/ after all the test methods in a test class

\*\*\*Bu notasyonlar kullanilince her method icin ayri ayri calismaz, soylendigi sekilde sadece bir kere calisir. (JUnitteki @BeforeClass ile ayni sekilde)

- @BeforeMethod @AfterMethod =>run before/ after each test method

\*\*\* Her methoddan once veya sonra calisir. (JUnit deki @Before method gibi)



# TestNG

## @Before @After Annotations

```
import org.testng.annotations.*;

public class TestNGFirst {
    @BeforeMethod @BeforeClass @BeforeTest @BeforeSuite//use only one
    public void setUp(){
        System.out.println("Contains what ever you need before executing a test" +
            "//such as driver set up, waits, maximize, and, go to main URL etc.");
    }

    @Test
    public void test1() { System.out.println("Test case 1"); }

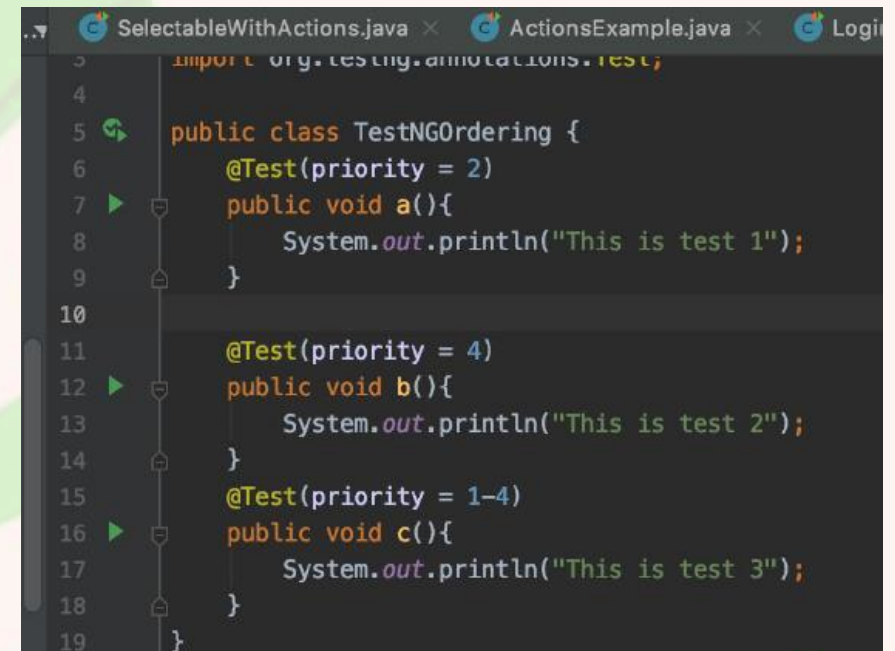
    @Test
    public void test2() { System.out.println("Test case 2"); }

    @AfterMethod @AfterClass @AfterTest @AfterSuite//use only one
    public void tearDown(){ System.out.println("Contains what ever you do at the end of the test case" +
        "such as closing/quitting driver, getting screenshots, generating reports, etc."); }
}
```

# TestNG

## Priority

- TestNG (default) olarak @Test methodlarını alfabetik sıraya göre run eder. (Yukardan asagi degil!)
- **priority** annotation Testlere öncelik vermek için kullanılır.  
Kucuk olan Numara daha once calisir.



```
import org.testng.annotations.Test;

public class TestNGOrdering {
    @Test(priority = 2)
    public void a(){
        System.out.println("This is test 1");
    }

    @Test(priority = 4)
    public void b(){
        System.out.println("This is test 2");
    }

    @Test(priority = 1-4)
    public void c(){
        System.out.println("This is test 3");
    }
}
```

# TestNG

## Priority

- 1) Bir class oluşturun: YoutubeAssertions
- 2) <https://www.youtube.com> adresine gidin
- 3) Aşağıdaki adları kullanarak 3 test metodu oluşturun ve gerekli testleri yapın
  - o titleTest => Sayfa başlığının  
“YouTube” olduğunu test edin
  - o imageTest => YouTube resminin  
görüntülendiğini (isDisplayed()) test edin
  - o *Search Box 'in erişilebilir olduğunu test edin (isEnabled())*
    - o wrongTitleTest=> Sayfa başlığının “youtube” olmadığını doğrulayın

# TestNG

## Handle Dropdown

- Adim1: Dropdown menuyu herhangi bir locator ile locate edin.

`WebElement selectElement=driver.findElement(By.id("value of id"));`

- Adim 2: Yeni bir "Select" objesi olusturun ve daha once locate ettigimiz WebElement'i parameter olarak yeni objeye ekleyin

`Select options=new Select(selectElement);`

- Adim 3: Dropdown icin kullanilan 3 yontemden biriyle dropdown menusundeki elemanlardan istediginizi secin

`options.selectByIndex(1);`

# TestNG

## Handle Dropdown

Dropdown menüsündeki elementleri 3 şekilde seçebiliriz

1. Index kullanarak `selectByIndex();`
2. Değer kullanarak `selectByValue();`
3. Görünen değerini kullanarak `selectByVisibleText();`

İstenirse `getOptions();` methodu kullanılarak DropDown'daki tüm elementler bir listeye konabilir. `List<WebElement>`

# TestNG

## Handling Dropdown

- Bir class oluşturun: **DropDown**
- <https://the-internet.herokuapp.com/dropdown> adresine gidin.
  - 1.**Index** kullanarak Seçenek 1'i (Option 1) seçin ve yazdırın
  - 2.**Value** kullanarak Seçenek 2'yi (Option 2) seçin ve yazdırın
  - 3.**Visible Text**(Görünen metin) kullanarak Seçenek 1'i (Option 1) seçin ve yazdırın
  - 4.Tüm dropdown değerleri(value) yazdırın
  5. Dropdown'un boyutunu bulun, Dropdown'da 4 öge varsa konsolda **True** , değilse **False** yazdırın.