



# Selenium

## Page Object Model Framework design

18 ARALIK 2020  
DERS 21

## Driver Class

Mehmet BULUTLUOZ  
Elektronik Muh.

TECH PRO ED

---

# Page Object Model

## Driver Class

Temel Hedefimiz: Test methodlarımızda kullanacagimiz WebDriver driver'i utilities altindaki Driver Class'inda olusturmak ve testlerimizde bu class ismi uzerinden driver'a ulasip olusturma,kullanma ve kapatma islemlerini yapmak.

```
public class Driver {  
  
    static WebDriver driver;//IQ:Why WebDriver static  
  
    public static WebDriver getDriver(){  
        if (driver==null){//if driver is not pointing  
            WebDriverManager.chromedriver().setup();  
            driver=new ChromeDriver();  
        }  
        return driver;  
    }  
}
```

Basic Driver Class

```
public class SingletonDriver {  
  
    @Test  
    public void test(){  
        Driver.getDriver().get("https://www.google.com/");  
    }  
}
```

Test Class(driver->Driver.getDriver())

---

# Page Object Model Driver Class

## Singleton Pattern(Tekli Kullanım)

- Herhangi bir Java classında yalnızca bir obje oluşturmamıza izin veriliyorsa, bu tür classlara singleton class denir. Yani singleton pattern(tekli kullanım) bir Class'ı tek bir instance ile kısıtlayan bir software dizayn kalıbıdır.
  - Genel olarak aynı gereksinimimiz olduğunda, ve buna her ihtiyaç duyulduğunda ayrı bir obje oluşturulması tercih edilmez. Singleton class; kullanıcıyı Webdriver'ın gerekli olduğu tüm örnekler(instance) için aynı objeyi kullanmaya zorlar.
  - Sadece bir obje oluşturmali ve buna her ihtiyaç duyduğumuzda kullanmalıyız. Bu performans ve bellek kullanımı için de faydalıdır.
  - Tüm framework'te veya tüm classlarda koordine için tek bir obje gerektiğinde faydalıdır.
-



# Page Object Model

## Driver Class

### Tüm Browserlar için Kullanım

- Driver Classımızı tüm tarayıcılar(browser) için geliştireceğiz.
- Driver objesi create etmeden önce farklı tarayıcı koşullarını kontrol etmek için switch statement kullanıyoruz.
- Sonra TestBasede yaptığımız gibi “wait” koyabiliriz.
- Ardından driver’i kapatıyoruz.
- Şu andan itibaren TestBase Classını değil Driver Classını kullanacağız.

```
public class Driver {  
  
    private Driver(){//so we cannot create another object. So we can have sing  
    }  
  
    static WebDriver driver;//IQ:Why WebDriver static? Our utilities are stati  
    |  
    public static WebDriver getDriver(){  
        if (driver==null){//if driver is not pointing nothing then assign Chro  
            switch (ConfigReader.getProperty("browser")){//We will get the bro  
                case "chrome":  
                    WebDriverManager.chromedriver().setup();  
                    driver=new ChromeDriver();  
                    break;  
                case "firefox":  
                    WebDriverManager.firefoxdriver().setup();  
                    driver=new FirefoxDriver();  
                    break;  
                case "ie":  
                    WebDriverManager.iedriver().setup();  
                    driver=new InternetExplorerDriver();  
                    break;  
            }  
        }  
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  
        driver.manage().timeouts().pageLoadTimeout(10, TimeUnit.SECONDS);  
        driver.manage().window().maximize();  
        return driver;  
    }  
  
    public static void closeDriver(){//After we are done with the driver we cl  
        if (driver!=null){//quick when driver is pointing chrome, firefox, etc  
            driver.quit();  
            driver=null;//setting back to null, so making sure driver set to n
```

---

# Page Object Model

## POM Class Work

### Global Trader Sign Up Test.

Kullanici <https://www.glbtrader.com/register.html> adresine gitsin.

1. **pages** paketinin altina bir page class olusturun : **GlbSignUpPage** ve Page objelerini(webelement) locate edin.

2. **smoketest** paketinin altina bir test classi olusturun : **GlbSignUpTest**

Kullanici kimlik bilgileri ile kayit oldugunda “**Success!**” mesajini gordugunu test edin

TECHPROED

---

---

# Page Object Model

## POM Class Work

### Global Trader Sign In Test.

Kullanici <https://www.glbtrader.com/register.html> adresine gitsin.

1. **pages** paketinin altina bir page class olusturun : **GlbHomePage** ve Page objelerini(webelement) locate edin.
2. **smoketest** paketinin altina iki tane test classi olusturun :
  - **PositiveTestGlbSignIn** : valid kullanıcı adı ile giriş yapabildigimizi test edin
  - **NegativeTestGlbSignIn** : 3 farklı senaryo ile yanlış şifre veya kullanıcı adı ile sayfaya giriş yapılamadigini test edin