



**BATCH** : Batch 59  
**LESSON** : **SELENIUM 28**  
**DATE** : 10.06.2022  
**SUBJECT** : Interview Prep.1



techproeducation



techproeducation



techproeducation



techproeducation



techproedu



# Interview Prep.

1- Selenium'un avantajlari ve dezavantajlari nelerdir ? Selenium'da hangi versiyonda calistiniz?

- Selenium nedir ? : Browser'lari otomate eden tool'larin calismasi icin suite'dir. Selenium browser'lari otomate eder, bu guc ile ne yapacaginiz size kalmis

Avantajlari :

- Acik kaynaklidir
- Bir cok browser icin destek verir
- Farkli programlama dileri ile calisir
- Tum isletim sistemleri ile uyumlu calisir
- Acik kaynakli oldugu icin profesyonel bir musteri hizmetleri yoktur ama cok genis bir kullanici kitlesi ve bunlari bilgi paylasim portallari mevcuttur

Dezavantajlari :

- Programlama dili bilmeniz gerekir
- Sadece web tabanli uygulamalar ile calisir. (Appium ile birlikte mobil testing yapılabilir)
- Performans testleri yapmaz, daha cok functional testlere odaklanir
- Profesyonel bir musteri hizmetleri yoktur
- Handle captcha mumkun degildir



# Interview Prep.

2- Selenium'da expected exceptions nedir ?

- Eger bir test method'u icerisinde yazdigimiz kodlar exception olusturma ihtimali tasiyorsa, test method'unun yanina throws keyword ile bu exception' lari yazariz

FileNotFoundException

FileNotFoundException

Thread.sleep



# Interview Prep.

3- POM (Page Object Model) nedir ? Avantajlari nelerdir?

- POM bir framework design pattern'dir (pom.xml ile karistirilmayalim)

Java'daki OOP konsept cercevesinde farkli package ve class'larda olusturulan variable, method ve test Data'larini yoneterek daha kullanisli, maintenance'li kolay ve reusable method'lar sayesinde daha az kod yazarak daha cok is yapabildigimiz bir framework amaclar.

1.Package : pages : locate'ler ve varsa o sayfa ile ilgili reusable methods

2.Package : utilities : testlerimizin disinda olup testlerimiz icin mutlaka olmasi gereken class'lari koydugumuz package'dir (ihtiyac duyulan arac gerek).

**Driver** : Normalde biz selenium WebDriver class'indan driver olusturarak tum islemlerimizi yapabiliyorduk. Ancak bunlari tekrar tekrar olusturmak, tam sayfa yapmak, implicitly wait eklemek gibi islemleri tekrar tekrar yapmamak icin Driver class'ini olusturup buradaki getDriver() ve closeDriver() method'lari ile tum islemleri tek seferde yapmis olduk.

Bu method'lari static yaparak her seferinde obje olusturma ihtiyacinin onune gectik

Driver class'ini olustururken configuration.properties'e yazacagimiz browser ismine gore calismasi icin bir switch yapisi icerisinde kullanabilecegimiz tum browser'lar icin gerekli ayarlamalari yaptik.

Driver class'i sayesinde TestBase bagimlilikinden kurtulduk.



# Interview Prep.

## 3- POM (Page Object Model) nedir ? Avantajlari nelerdir?

TestBase : İlk basta her test class'inda driver olusturmak ve ilgili ayarlari yapmak zorunda kaliyorduk. OOP concept'teki inheritance ozelligini kullanarak her class'da bunlari yapmak yerine, istedigimiz class'lari TestBase'e child yaparak bu method'lari otomatik olarak kullanma hakkimiz oldu. Her seferinde TestBase'e child yapmak baska inheritance yapmamiza imkan birakmiyordu. Bu yuzden Driver class'ini olusturarak OOP'nin class adi ile static method cagirma ozelligini kullanarak TestBase'den kurtulduk.

ConfigReader: configuration.properties dosyasinda bulunan TEST DATALARINI okuyabilmemiz ve kullanabilmemiz icin meydana gelen class'dir.

### 3.Package : Tests:

POM'de kullandigimiz diger package ve dosyalar sayesinde Tests package'indaki testlerimiz tamamen dinamik halde olur. Yani manuel olarak yazdigimiz hicbir test datasi veya locator olmaz.

configuration.properties file: Test datalarimizi tuttugumuz file'dir.

key-value ikileleri seklinde calisir.

key'ler unique olmak zorundadir. Cunku bu key ile value'lari getirip testlerimizde kullaniyoruz. Duplicate olursa hangi degeri kullanacagini bilemeyiz



# Interview Prep.

## 4- Page Factory nedir?

- POM'in temel taslarindan page class'ini olusturur olusturmaz, bir Constructor olusturup PageFactory class'indan `initElements()` method'u kullanarak driver'imizi page sayfasinda kullanilir hale getirmemizdir.



# Interview Prep.

5- Selenium'da senkronizasyon deyince ne anliyorsunuz ? Senkronizasyon nasıl sağlanır ? Hangi method'lar kullanılır?

- Senkronizasyon : Testlerimizi çalıştırmak için kullandığımız driver objesi ile browser'imizin uyumlu bir şekilde çalışmasını sağlamaya denir.

Driver kod ortamında çalıştığından çok hızlı ilerler ancak browser. Bilgisayar, internet, hedef web uygulaması gibi değişkenlerin hızlarına bağlı olarak ilerleyebilir.

Bu durumda testimiz ilerlerken driver ile browser arasında kopmalar olabilir.

Bu kopmaların olmaması için gerektiğinde kodlarımızı wait yöntemleriyle BEKLETİRİZ

1- Hard wait : Thread.sleep() : Java'dan gelir, kendisine verilen süre kadar kodların çalışmasını durdurur. Thread.sleep'in olduğu satırdan bir sonraki satıra geçiş sadece ve sadece kendisine tanımlanan sürenin bitmesi ile mümkündür.

2- Dinamik wait : selenium'dan gelir ve tanımlanan görevi bitirirse sürenin dolmasını beklemeyiz, kodu çalıştırmaya devam eder.

Eğer tanımlanan süre içerisinde görev tamamlanamazsa o zaman exception fırlatır.



# Interview Prep.

5- Selenium'da senkronizasyon deyince ne anliyorsunuz ? Senkronizasyon nasıl sağlanır ? Hangi method'lar kullanılır?

İki çeşit dinamik wait kullanıyoruz

1- `implicitlyWait( Duration.ofSeconds(15))` : sayfanın yüklenmesi veya web elementlerin locate edilmesi için global bir zaman belirler.

sayfa yüklenmesi ve her bir web elementin bulunması için kendisine tanımlanan süre kadar bekleyebilir.

Bu süre zarfında görev tamamlanmazsa exception verir

2- `ExplicitlyWait()` : genel bir bekleme süresi değil spesifik bir web element için tanımlanmış, spesifik bir görev için beklenecek max süreyi belirler

Bunun için ilgili webElementin locate'i ve wait objesine ihtiyacımız vardır.

Eğer kullanılacak Webelement başlangıçta html kodlarında bulunamıyorsa, locate ve bekleme aynı satırda yapılabilir

```
wait.until(ExpectedConditions.visibilityOf(HerakuappPage.deleteButton));
```

```
WebElement element = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("...")));
```





# Interview Prep.

6- Bir objenin gorunur olmasini 1 dakika beklememiz gerekirse bunu nasil yaparsiniz ?



```
WebDriverWait wait= new WebDriverWait(driver,Duration.ofSeconds(60));  
WebElement element = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("...")));
```



# Interview Prep.

8- JavaScript executor nedir? Projenizde nerede kullanmистiniz?

- Bazen driver objemizle cozedemedigimiz otomasyon islemleri icin JavaScriptExecutor kullaniriz.  
Java Script light Weight bir programdir  
Yani Java Script ile yazilan bir kod parcacigi (script) nereye konursa calisir.  
Biz de java Script'te tanimlanmis hazır fonksiyonlari kullanabiliriz

1- JavaScriptExcetor objesi olusturmaliyiz

JavascriptExecutor js = (JavascriptExecutor) driver; (var olan driver'ı java script'e cast ediyoruz)

2- JavaScriptExcetor icin tanimlanmis hazır fonksiyonlari kullaniyoruz

js.executeScript ( "ilgili script" , web element); genel kullanimi

jse.executeScript("arguments[0].scrollIntoView(true);",webtablePage.nameOfList)



# Interview Prep.

9- Selenium'da mouse hareketleri ve Klavye fonksiyonlarını nasıl otomate edersiniz ?

- Selenium'da klavye ve mouse ile yaptığımız tüm işlemleri Actions class'ı ile otomate ederiz.

- actions objesi oluştur

```
Actions actions= new Actions( Driver.getDriver( ));
```

- kullanacağımız tüm web elementlerini locate edip bir WebElement'e atayalım

- actions.....

```
.....
```

```
..... perform()
```

yapacağımız tüm işlemleri yapıp sonra perform diyerek actions objesini aktif ediyoruz.

Actions method'ları mouse base ve klavye base olarak ikiye ayrılır.

Klavyede çok fazla tuş olduğundan tüm tuşlar için ayrı ayrı method'lar yazmak yerine

- tek seferlik kullanım için `sendKeys( Keys.ENTER)`

- uzun süreli basılı tutma işlemi için `keyDown(Keys.SHIFT)` , isimiz bitince elimizi tuştan kaldırmak için ise `keyUp( Keys.SHIFT)`



# Interview Prep.

9- Selenium'da mouse hareketleri ve Klavye fonksiyonlarini nasil otomate edersiniz ?

Mouse base Actions

`click()`

`dragAndDrop()`

`moveToElement()`

`contextClick()`

`doubleClick()`

`clickAndHold()`



# Interview Prep.

10- Önceden locate edilmiş bir element'in her hangi bir attribute'unun değerini nasıl alırsınız ?

- `WebElement playButonu= driver.findElement(By.id("playButonu"));`

Name attribute'unun değeri nedir ?

```
String nameValue= playButonu.getAttribute("name");
```



# IntelliJ IDEA NEDİR

11- Selenium ile pop-ups'leri handle edebilir misiniz ?

- Selenium ile tum pop-up'leri handle edemeyebiliriz. Ancak HTML ve JS alert'leri handle edebiliriz.

HTML alert : HTML elementi oldugu icin locate edilebilir ve kullanılabilir

JS Alert'ler

`driver.switchTo().alert()` Dedikten sonra var olan method'lardan birini kullanabiliriz

`.getText()` : alert'in uzerinde bir yazi varsa bize dondurur

`.accept()` : OK butonuna basmis oluruz

`.dismiss()` : cancel butonuna basmis oluruz

`.sendKeys("istene Yazi");` : eger alert prompt ise gonderdigimiz yaziyi data olarak Kabul eder ve alert'un icerisine yazar



# IntelliJ IDEA NEDİR

12- Selenium'da multiple windows'u handle edebilir misiniz ? Nasıl ?

- Selenium 4 multiple window konusunda yenilik getirdi

Daha önceden bizim istegimizle yeni bir window veya tab acilmazken, Selenium 4 bu ozelligi getirdi.

- Multiple window handle edilirken, yeni penceri biz newWindow diyerek actik ise , o zaman driver otomatik olarak yeni windows'a gecer.

Bu tur calismalarda yeni windows'a gectikten sonra eski pencereye donmeniz gerekecekse daha ilk window'da iken onun windowhandle degerini alip kaydetmeliyiz

- Eger bir link tikladigimiz icin yeni window aciliyorsa, driver yeni window'a gecmez bu durumda bizim Java kullanarak tum window'larin handle degerlerini elde etmemiz gerekir.

- ilk sayfada iken ilkSayfahandle degerini alip kaydediyoruz

- ikinci sayfa acildiktikten sonra getWindowHandles( ) method'u ile acik tum window'larin windowHandle degerlerini alip bir Set'e atiyoruz. Bu set'deki 2 handle degerini inceleyerek ikinciSayfaHandle degerini bulup bir variable'a atiyoruz.



# IntelliJ IDEA NEDİR

13- Selenium'da Assert ile Verify komutlarini aciklar misiniz ?

- Assert ile testimizi yaptigimizda sonuc istenmeyen sekilde cikarsa execution durur ve hata mesaji yayınlanır
- Verify ise execution'ı durdurmadan istenen dogrulamanin yapilip yapilmadigini bize rapor eder.

Genelde verify islemi icin Soft Assert kullanilir.

Soft Assert'de obje olusturma baslangic noktası, assertAll ise bitis noktasidir.

Soft assert bitis noktasina kadar tum assertion'lari yapip sonuclari kendi icerisinde tutar.

Bitis noktasina vardiginda, yaptigi tum assertion'lari raporlar ve failed olan varsa assertAll satirinda execution durur

Tum assertion'lar Passed olduysa execution durmaz testimiz calismaya devam eder





# IntelliJ IDEA NEDİR

14- Selenium'da screenshot alabilir misiniz ?

- Selenium 3'te sadece tum sayfanin screenshot'l alinabilirken, selenium 4 ile istedigimiz web elementlerin de spesifik screenshot alma ozelligi geldi

Tum sayfanin screenshot'ini almak icin

- 1- TakesScreenshot scrShot =((TakesScreenshot)webdriver); **obje olustur**
- 2- screenshot'l gecici olarak kaydet `File geciciDosya = ts.getScreenshotAs(OutputType.FILE);`
- 3- asil dosyayi olustur `File tumSayfaResim = new File("target/ekranGoruntuleri/tumSayfa" + tarih + ".jpeg");`
- 4- gecici dosyayi, asil dosyaya kaydet `FileUtils.copyFile(geciciDosya, tumSayfaResim);`

Spesifik bir webelementin screenshot'ini alacaksak

- 1- locate yap
- 2.Adimi locate yaptigin webelement uzerinden calistir.



**BATCH** : Batch 59  
**LESSON** : **SELENIUM 29**  
**DATE** : 11.06.2022  
**SUBJECT** : Interview Prep.2



techproeducation



techproeducation



techproeducation



techproeducation



techproedu



# IntelliJ IDEA NEDİR

**IntelliJ** : ide : yazdigimiz java kodlarinin vcalismasi icin bir copiler'dir. Java kodlarinizin calismasi icin mutlaka bir compiler kullanmalisiniz. Selenium ide, eclipse veya farkli ide'ler de kullanılabilir  
Java : Otomasyonda kullandigimiz kodlarin calistigi programlama dili.

**Selenium** : Selenium browser'lari otomate etmek icin kullanacagimiz farkli tool'larin calisabilecegi bir suite'dir. Selenium icin ihtiyacimiz olan kutuphaneleri istersek elle selenium sitesinden indirip, Java'daki `System.setProperty("webdriver.istenenBrowser.driver","driver'in dosya yolu");` kodu ile projemize ekleyebiliriz.  
Veya Maven gibi bir build tool kullanirsak oradaki kutuphane ekleme yontemini kullanabiliriz

**Maven** : build and management tool. Selenium ile kullandigimiz bir projede ihtiyacimiz olan tum kutuphaneleri kolayca projeye eklemek ve ihtiyacimiz oldugunda kolayca update etmek icin kullandigimiz bir tool'dur. Maven'in getirdigi standartlar cercevesinde tum tool'lar GitHub'a ekledikleri kutuphanelerini dependency denen kucuk kod bloklarini kullanarak projemize ekleyebiliriz.

**pom.xml** : maven proje'nin kalbidir. Dependency ve plug-inn'leri ekledigimiz xml dosyasidir.

**framework** : Bir projede ortak olarak calisan tum paydaslarin, projedeki dosya ve datalari rahatlikla anlamasi, gelistirmesi, update etmesi ve maintenance edebilmesi icin kabul edilmis ortak proje duzenidir.



# IntelliJ IDEA NEDİR

15- Selenium'da browser'in boyutunu ve konumunu istedigimiz deger olarak ayarlayabilir misiniz ?

- Evet `driver.manage().window().get` ve `set` method'lari ile browser'in boyutlarini ve konumunu pixel olarak alabilir, ve istenirse boyut ve pozisyon istenen degere getirilebilir.

Bunun disinda `maximize()`, `minimize()`, `fullscreen()` gibi method'larla window boyutunu default olarak tanimlanmis boyutlara getirebilirsiniz



# IntelliJ IDEA NEDİR

16- Selenium'da dropdown menu'yu nasıl handle edersiniz ? Dropdown'da var olan bir değeri nasıl seçersiniz ?

- Selenium'da dropdown denince akla select gelir. HTML kodları da select tag'i ile başlar.  
Bir dropdown 3 adımda handle edilebilir

1- dropdown webelementi locate edilir

2- Select select= new Select(locateEdilenDropdownWebelementi)

3- Dropdownda istediğimiz işlemi yapmak

- eğer bir option seçmek istersek 3 method kullanabiliriz  
    visibleText( ), index( ), value( )
- son seçilen option'ı yazdırma  
    getFirstSelectedOption( )
- tüm opsiyonları almak için  
    List<WebElement> opsiyonlar= select.getOptions( );



# IntelliJ IDEA NEDİR

17- Selenium'da bir web uygulamasında var olan uploadFile butonunu kullanarak bilgisayarınızdan bir dosyayı nasıl uygulamaya upload edersiniz ?

- Selenium Webdriver üzerinden web uygulamalarına ulaşır ama bilgisayarımızdaki dosyalara erişemez. Eğer bilgisayarımızdaki bir dosyaya ulaşip dosyadaki bilgileri class'imiza getirmek istiyorsak dosyaya erişim kısmını java ile halletmeliyiz.

```
FileInputStream fis= new FileInputStream(dosyaYolu);
```

dosya yolu her bilgisayarda farklı olur. Çünkü bilgisayar adı ve kullanılan user ismi bilgisayardan bilgisayara değişir. Biz kodlarımızı dinamik yapmak istersek dosya yolunu iki parçaya ayırıyorduk her bilgisayarda değişen kısım bilgisayarAdi//userAdi// → getProperty("user.home") bundan sonrası tüm bilgisayarlarda aynı olur user.home // Desktop gibi...

- Dosya upload için ise dosya yolunu dinamik olarak hazırlamak yeterli olacaktır.
  - 1- uygulamadaki dosyaSec butonunu locate edeceğiz
  - 2- dosyaSecButonu.sendKeys(dosyaYolu)
  - 3- uploadButonu.click( );



# IntelliJ IDEA NEDİR

## 18- WebDriver nedir?

- interface'dir.

Bizim elelerimizle, klavye ve mouse kullanarak yaptığımız tüm işlemleri ve gözümüzle yaptığımız kontrolleri bizim adımıza yapan nesneyi üretmek için kullanırız.

madem WebDriver bir interface'dir nasıl nesne üretebilirsiniz ?

evet WebDriver bir interface'dir ancak onun özelliklerini inherit eden tüm browser'ların class'ları vardır. Biz bir driver oluşturmak istediğimizde data turu olarak WebDriver interface'ini, constructor olarak ise istediğimiz browser'ın class'ını kullanıyoruz.

overriding sayesinde nesne ile kullandığımız tüm method'lar constructor'ın olduğu class'dan çalışır.

```
WebDriver driver= new ChromeDriver();
```

```
driver.get()
```



# IntelliJ IDEA NEDİR

19- Bir web sayfasinin kaynak kodlarini nasil alabilirsiniz ?

- driver method'larinda bulunan `driver.getPageSource( )` method'u ile sayfa kaynak kodlarini class'imiza getirebiliriz.





# IntelliJ IDEA NEDİR

20- driver.get( url ) ile driver.navigate( ).to(url)' in farklari nelerdir ?

- navigate ile gittigimiz sayfalarda navigate method'larine kullanabiliriz

driver.navigate( ).to(url)

driver. navigate( ).back( ); bir onceki sayfaya doner

driver. navigate( ).forward( ); back( ) ile dondugumuz bir sayfadan tekrar ileriye gider

driver.navigate( ).refresh( ); sayfayi yeniler



# IntelliJ IDEA NEDİR

## 22- Maven nedir ? Avantajlari nelerdir ?

- **Maven** : build and management tool. Selenium ile kullandigimiz bir projede ihtiyacimiz olan tum kutuphaneleri kolayca projeye eklemek ve ihtiyacimiz oldugunda kolayca update etmek icin kullandigimiz bir tool'dur. Maven'in getirdigi standartlar cercevesinde tum tool'lar GitHub'a ekledikleri kutuphanelerini dependency denenen kucuk kod bloklarini kullanarak projemize ekleyebiliriz.  
**pom.xml** : maven proje'nin kalbidir. Dependency ve plug-inn'leri ekledigimiz xml dosyasidir.

Avantajlari : Bir projede calisan farkli gruplari senkronize olarak calismaya devam etmesini saglamak amaciyla baslamistir.

- Anlasilabilirlik
- Kolay guncelleme
- Surdurulebilirlik
- Acik kaynak
- farkli ide'lerle calisabilme
- projeler icin bir standard olusturma
- reusability



# IntelliJ IDEA NEDİR

23- dryRun nedir ? Nasıl kullandınız ?

- dryRun Runner'da kullandığımız parametrelerden biridir.

iki değer alabilir

<code>dryRun=false</code>	default değerdir, Runner'ın testlerimizi çalıştırması için bu değer seçilmiş olmalıdır
<code>dryRun= true</code>	testlerimizi çalıştırmadan sadece eksik adım olup olmadığını kontrol eder ve eksik adımların method'larına bizim için oluşturur

true değeri seçildiğinde test scenario'su failed olacak şekilde bile olsa, eksik adım yoksa test passed yazar



# IntelliJ IDEA NEDİR

24- Maven dependency nedir ? Nasıl kullandınız ?

- |



25- Bir listedeki tekrarları nasıl silersin ?

- |



# IntelliJ IDEA NEDİR

26- Dependency ve plugins arasında nasıl bir fark vardır ?

- |



# IntelliJ IDEA NEDİR

27- findElement( ) ile findElements( ) arasindaki farlar nelerdir ?

- |



# IntelliJ IDEA NEDİR

28- Bir webelement'te yazilmis olan bir text'i nasil silersiniz?

- |





29- `getText()` ile `getAttribute()` arasindaki far nedir ?

- |



30- Bir dropdown'da bulunan option'ların sirali olup olmadigini nasıl kontrol edersin ?

- |



31- Kac cesit locator vardir ? Hangisini tercih edersin ?

- |



# IntelliJ IDEA NEDİR

32- Absolute ve relative xpath arasındaki farklar nelerdir ?

- |



## 33- Behaviour Driven Development nedir ?

- |