



COPYRIGHT MATERIALS

TechPro Education
Copyright Material
For only TechPro Education Students



Selenium giriş

**SELENIUM GIRIS
OTOMASYON TEST NEDİR?
SELENIUM PROJESİ OLUSTURMA**



SELENIUM NEDİR

- Selenium, web uygulamalarını farklı tarayıcılarda ve platformlarda test etmek için ücretsiz (açık kaynaklı) bir araçtır.
- Selenium yalnızca web tabanlı uygulamaları otomasyon yapmaya odaklanır. Mobil ve Windows testi yapmak için eklentiler selenium'a eklenebilir.
- Selenium jar dosyaları ile kurulabilir. Kurulum sırasında jar dosyalarını gördünüz.

Selenium automates browsers. That's it!

What you do with that power is entirely up to you.

Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that.

Boring web-based administration tasks can (and should) also be automated as well.

- 2021'de piyasaya sürülen Selenium 4'ü öğreneceğiz.
- Selenium'u çeşitli programlama dilleri ile yazabilirsiniz java, Python, ruby, .Net vb.

About Selenium

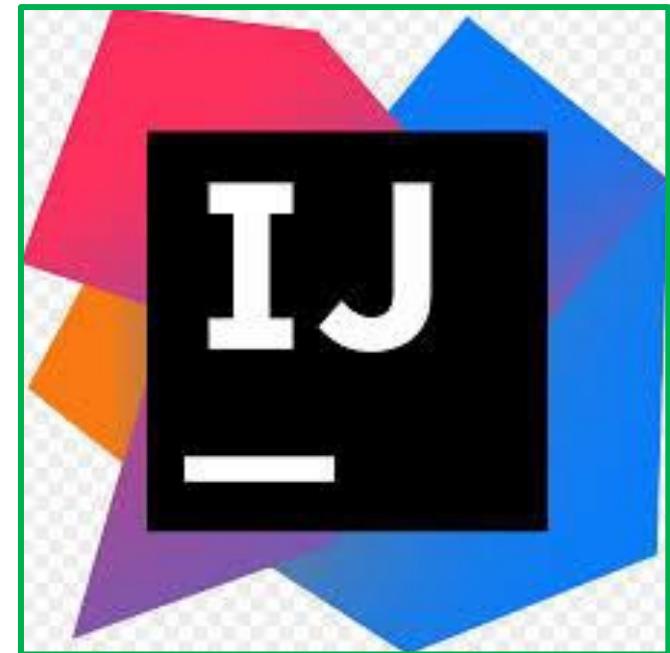
Selenium is a suite of tools for automating web browsers.

- Selenium, otomasyon yapmak için kendi sınıflarına ve yöntemlerine sahip bir suite'dir..



IntelliJ IDEA NEDİR

- IntelliJ IDEA 2000 yılında kurulmuş olan JetBrains isimli yazılım firmasına ait olan, popüler bir Java editörüdür.. (ide)
- IntelliJ IDEA'nın her yönü, geliştirici üretkenliğini en üst düzeye çıkarmak için tasarlanmıştır.
- Akıllı kodlama yardımcı ve ergonomik tasarım ile, kod yazımını yalnızca verimli değil, aynı zamanda keyifli hale getirir.
- Banyak framework ve plugin ile çalışma imkanı verir
- Akıllı tamamlama özelliği ile kod yazımını oldukça kolaylaştırır
- IDE ihtiyaçlarınızı tahmin eder ve sıkıcı ve tekrarlayan geliştirme görevlerini otomatikleştirir, böylece büyük resme odaklanabilirsiniz.





SOFTWARE TESTING NEDİR ?

EXPECTED RESULT (beklenen sonucun), **ACTUAL RESULT** (gerçek sonuca) esit olup olmadigini kontrol etme islemidir.

- Eger Expected result = Actual result, ise status PASSED (test basarili)
- Eger Expected result !=Actual result, ise status FAIL (test basarisiz)

Sonuç olarak, olması gereken şeylerin olmadığını veya olmaması gereken şeylerin olduğunu kontrol etmek ve ortaya çıkartmak yazılım testinin amacı olmalıdır.



- Her User Story icin Positive ve Negative Test(ler) yapılmalıdır
- Test, musteri/isletme ihtiyaçlarını karşılamak için yapılır.
- Bir uygulamayı test etmek için onceden belirlenmis user storyler (kullanici hikayeleri) ve tanımlanmis acceptance criterias (kabul kriterleri) dikkate alınır.

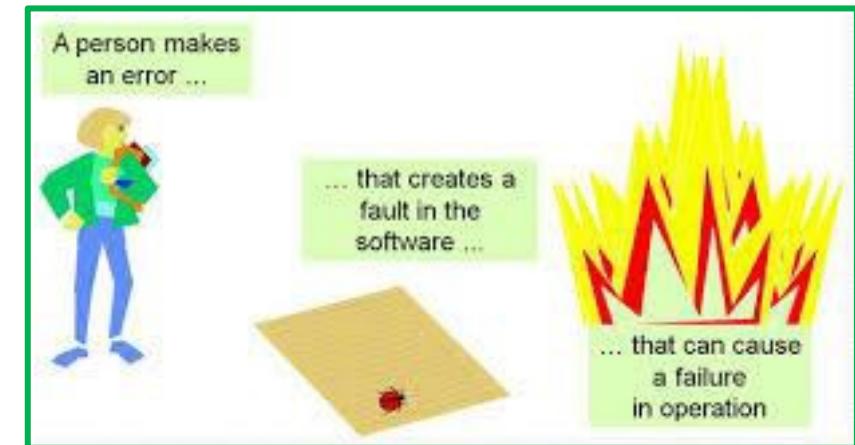


SOFTWARE TESTING NEDEN ONEMLIDIR ?

- İnsanlar hata yaparlar, bu hatalar kodda, yazılımda, sistemde ya da dokümanda defect (Kusur) oluşturur.
- Defect olan kod çalıştırıldığında sistem beklenen fonksiyonları gerçekleştiremez ve başarısız olur.

Bu sebeplerden dolayı;

- Müşteriye sunulmadan önce ürün kalitesinden emin olmak,
- Yeniden çalışma (düzelte) ve geliştirme masraflarını azaltmak,
- Geliştirme işleminin erken aşamalarında yanlışları saptayarak ileri aşamalara yayılmasını önlemek, böylece zaman ve maliyetten tasarruf sağlamak amacıyla ürün müşteriye sunulmadan önce test edilmesi gerekmektedir.



KISACA : Testing saves money and lives .

Testing para kayiplarini onler ve hayat kurtarir. (Ucak kazalari vs..)



MANUAL(FUNCTIONAL) TESTING NEDİR?

- Manuel test, uygulamayı herhangi bir otomasyon aracı olmadan manuel olarak test etmektir.
- Manuel test kullanıcıları dokümantasyon için sınırlı teknoloji (Excel vb.) kullanır, ancak otomasyon araçları veya dili kullanmazlar.
- Manuel teste insan hatası olabilir.
- Tüm Otomasyon Tester'lar, herhangi bir otomasyon yapmadan önce uygulamayı anlamak için mutlaka manuel test yapmalıdır.
- İyi bir otomasyon tester aynı zamanda iyi bir manuel testerdir.





TEST OTOMATION NEDİR ?



- Bir sistem bir otomasyon aracı (tool) yardımıyla test etmeye ‘Test Otomasyonu’ denir.
- Otomasyon test yazılımı test verilerini Test Edilen Sistem'e girebilir, beklenen ve gerçek sonuçları karşılaştırabilir ve ayrıntılı test raporları oluşturabilir.
- Bir test otomasyon tool'u kullanarak, calistirilan test paketini kaydetmek ve gerektiğinde yeniden calistirmak mümkündür. Test paketi otomatik hale getirildikten sonra hiçbir insan müdahalesi gerekmesizin programlandigi zamanda calisabilir.
- Giderek daha popüler hale gelmektedir.



MANUAL TESTING vs AUTOMATION TESTING

Asagida yazılı metin sizce neyi ifade etmektedir ?

- A. Test Case
- B. Manuel tester icin test adimlari
- C. Otomasyon ile test yapan kodlar

@hotel

Scenario: TC08 Positive login

```
Given kullanici "HMCUrl" sayfasina gider
And login linkine tiklar
Then kullanici adi olarak "HMCValidUsername" girer
And password olarak "HMCValidPassword" girer
And Login butonuna basar
Then basarili olarak giris yapildigini test eder
And sayfayı kapatır
```



MANUAL TESTING vs AUTOMATION TESTING

MANUAL TESTING

VS

AUTOMATION TESTING

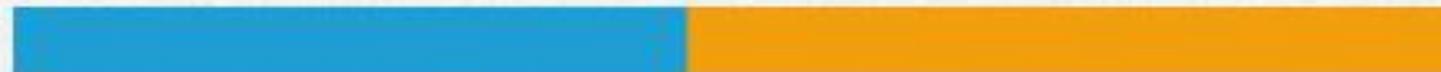
EXECUTION TIME



PEOPLE



INFRESTRUCTURE



TOOLS



TURNAROUND TIME

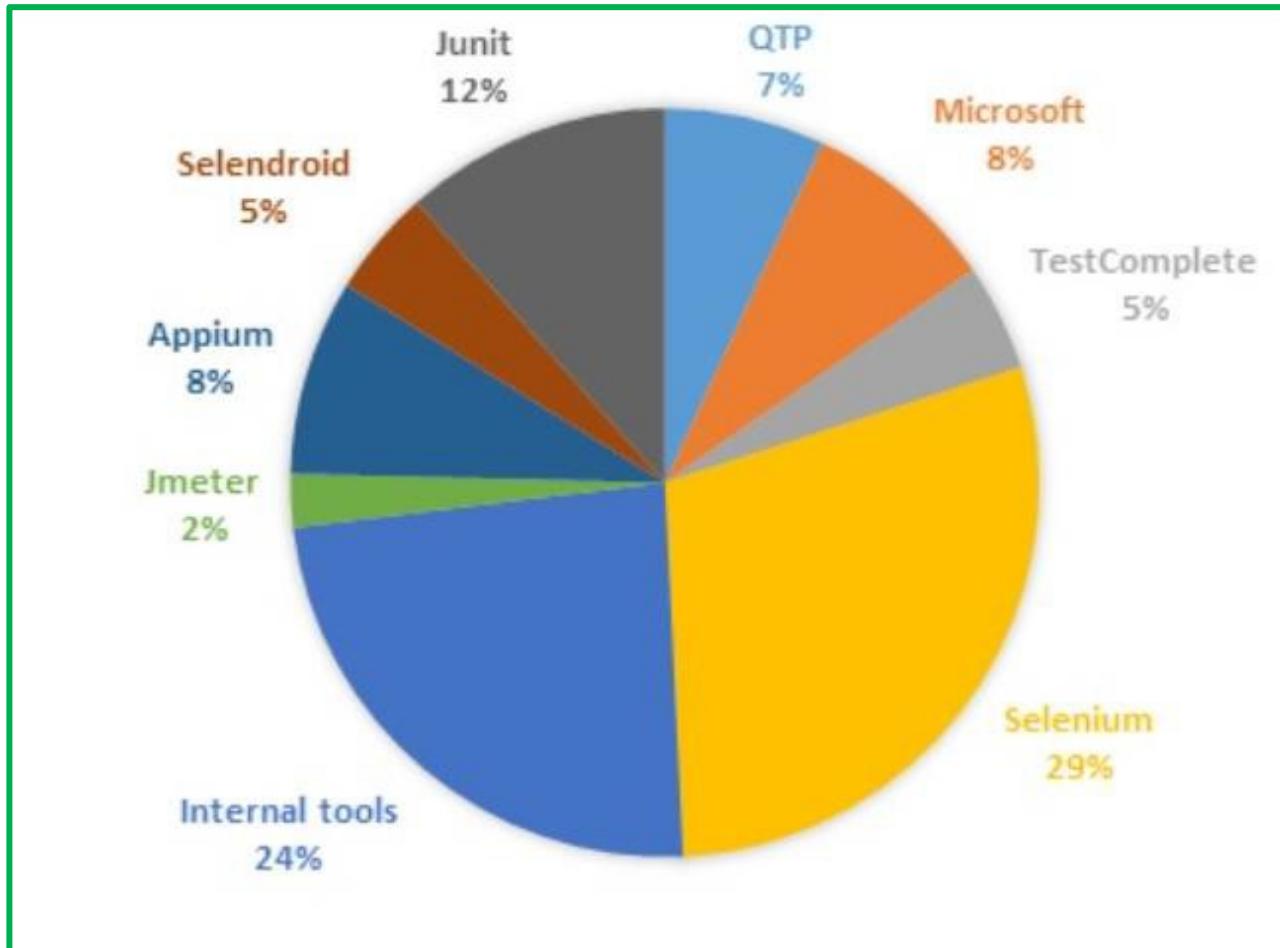


TRAINING





EN COK KULLANILAN TEST TOOL'LARI





Basarılı Bir Otomasyon Testi için Yapılması Gerekenler

- **Doğru otomasyon araçlarını(tools) seçin.** Genellikle şirketlerin kullandıkları tool'lari vardır, bazı şirketler de sizin isteginize gore tool secebirlirler.
- Uygulamanız hakkında **iyi bilgi sahibi olun.**
- **Test senaryolarınızı (test cases) kısa ve bağımsız tutun.**
- **Test otomasyonlarınızı önem derecesine göre sıralayın.**
- **Otomasyondan önce test verilerini hazırlayın(id,url,environment)**
- **Gerekirse, test caselerinizi yönetin(manage) ve bakımını yapın(maintain).** Yeni test caselerinizin eski otomasyon test scriptlerini bozmadigindan emin olun.
- **Testlerinizde hata olabilecegini goz onunde bulundurun.** Bir BUG'i rapor etmeden once mutlaka testinizi gozden gecirin
- **Her zaman test ekibinizle, özellikle team liderleri ile iyi iletisim kurun.**





SELENIUM BILESENLERİ

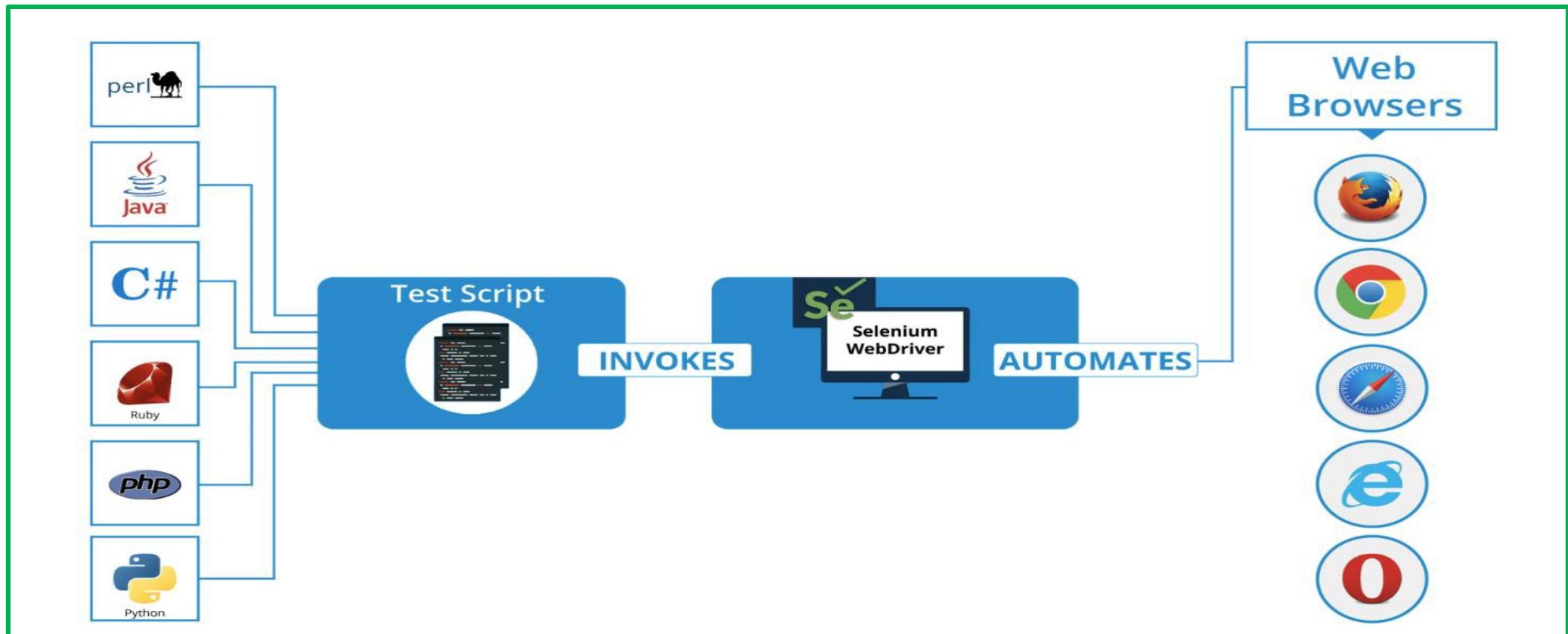


Selenium'un üç bileşeni vardır;

- Selenium Integrated Development Environment (IDE)
(Selenyum Entegre Geliştirme Ortamı (IDE))
- WebDriver (Biz Selenium WebDriver kullanacağız)
- Selenium Grid (paralel test için kullanılıyor)



SELENIUM NASIL CALISIR?





SELENIUM'UN AVANTAJLARI NELERDIR?



1.) Ücretsiz ve acik kaynaklidir.

(Open source)

2.) Bir çok programlama dilini destekler

(Java, Python, PHP, C#, Ruby vs.)

3.) Çoklu işletim sistemleriyle çalışır.

Multiple operating systems

(Windows, MacOS, Linux)

4.) Birden çok tarayıcı ile çalışır.

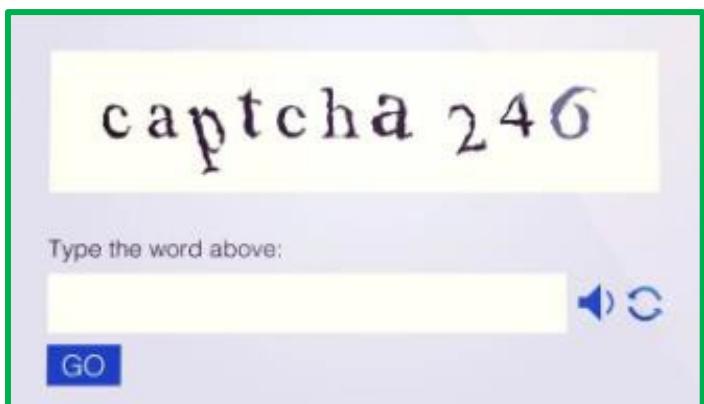
Multiple browsers

(IE, Safari, Chrome, Firefox vs.)



SELENIUM'UN DEZAVANTAJLARI NELERDIR?

- Programlama bilgisi gerektirir
(Biz Java biliyoruz)
- Yalnızca web tabanlı uygulamaları test eder
- Profesyonel desteğe sahip değil
(Ama geniş bir kullanıcı kitlesi var)



SELENIUM'UN YAPAMADIKLARI

- performans testi
- handle captcha
(diğer tüm otomasyon araçları gibi)



TEST AUTOMATION FRAMEWORK NEDİR?

- **Test Otomasyon Framework (TAF)**, yazılım projelerindeki Test Case'lerin oluşturulma ve gözden geçirme süresini azaltmak ve QA ekibimizin isini kolaylaştırmak için tasarlanmış yapılardır.

(İşinizi rahat yapabilmeniz için hazırlanmış çalışma ortamınızdır)
- **Kullanılan Tool'lar**
Java, Eclipse/
IntelliJ,
Selenium,
JUnit/
TestNG,
Cucumber
- **Test Yapılan Layer'lar**
UI , API , DataBase

The screenshot shows the IntelliJ IDEA interface with a Java project named "mymavenproject". The project structure is displayed in the left panel, showing the main directory, src (with main and test sub-directories), and resources. The test directory contains a java sub-directory with several test classes: CheckboxAndRadioBut, DropDownAmazon, DropDownHomework, FirstMavenClass, FirstTestNgClass, TestNGAnnotationsExar (which is currently selected), TestNGAnnotations, and WebTables. The right panel shows the code editor with the following Java code:

```
package com.techproed;  
  
public class TestNGAnnotationsExample {  
}
```



SELENIUM ve CHROME DRIVER KURULUMU

- 1) <https://www.selenium.dev/downloads/> adresine gidelim
- 2) Selenium Client & WebDriver Language Bindings altında Java driver'ini download edin
- 3) Browsers altında Chrome documentation linkini tiklayalim

Chrome'un kendi sayfasina gidip Current stable release'i tiklayip size uygun olani download edin
*** buradaki surum ile bilgisayarinizdaki Chrome surumunun ayni oldugundan emin olun
- 4) src altında resources director'si olusturun
- 5) Bu klasor altında drivers ve libraries klasorleri olusturun
- 6) Indirdigimiz chromedriver'i drivers klasorune, selenium-java dosyasini ise libraries klasorune cikartalim
- 7) intelliJ 'de yeni project / package / class olusturalim ve class icinde main method olusturalim
- 8) File/Project Structure/Modules/Dependencies kismindan jar dosyalarini yukleyelim



WebDriver Olusturulmasi

1. Yeni bir class oluşturun: class name ⇒ C01_Get

2. main method oluşturun

3. Java'dan System.setProperty("", "") methodunu kullanarak webdriver'in turunu ve path'ini belirleyelim.

`System.setProperty("webdriver.chrome.driver", "src/driver/chromedriver"); //MAC`

`System.setProperty("webdriver.chrome.driver", "src/driver/chromedriver.exe"); //WINDOWS`

4. Chrome driver oluşturun

`WebDriver driver = new ChromeDriver();`

5. Olusturdugumuz driver objesi ile WebDriver class'indan static method'lari kullanın



driver.get(); Method'ları

driver.get(String Url); String olarak girilen Url'e gider

driver.getTitle(); Icinde olunan sayfanın başlığını
String
olarak getirir

driver.getCurrentUrl(); Icinde olunan sayfanın Url'ini
String olarak getirir

driver.getPageSource(); Icinde olunan sayfanın
kaynak
kodlarını String olarak getirir

driver.getWindowHandle(); Icinde olunan sayfa ve/veya tab'ların
driver.getWindowHandles(); handle değerlerini getirir



driver.get(); Method'lari

Class Work

1. Yeni bir package olusturalim : day01
2. Yeni bir class olusturalim : C03_GetMethods
3. Amazon sayfasina gidelim. <https://www.amazon.com/>
4. Sayfa basligini(title) yazdirin
5. Sayfa basliginin “Amazon” icerdigini test edin
6. Sayfa adresini(url) yazdirin
7. Sayfa url’inin “amazon” icerdigini test edin.
8. Sayfa handle degerini yazdirin
9. Sayfa HTML kodlarinda “Gateway” kelimesi gectigini test edin
10. Sayfayı kapatın.



driver.navigate(); Method'ları

driver.navigate().to(String Url); String olarak girilen Url'e gider

driver.navigate().back (); Icinde olunan sayfadan, geldigi onceki sayfaya dondurur

driver.navigate().forward (); Back ile donulen bir sayfadan tekrar ileri gider

driver.navigate().refresh (); Icinde olunan sayfayı yeniler.



Selenium Navigation Methods

Class Work

1. Yeni bir Class olusturalim.C05_NavigationMethods
2. Youtube ana sayfasina gidelim . <https://www.youtube.com/>
3. Amazon soyfasina gidelim. <https://www.amazon.com/>
4. Tekrar YouTube'sayfasina donelim
5. Yeniden Amazon sayfasina gidelim
6. Sayfayı Refresh(yenile) yapalim
7. Sayfayı kapatalim / Tüm sayfaları kapatalim



driver.manage().window(); Method'lari

driver.manage().window().maximize(); Browser'i maximize yapar

driver.manage().window().minimize(); Browser'i minimize yapar

driver.manage().window().fullscreen(); Browser'i full screen yapar

driver.manage().window().getPosition(); Browser'in koordinatlarini verir

driver.manage().window().getSize(); Browser'in olculerini verir



Selenium Navigation Methods

Class Work

1. Yeni bir Class olusturalim.C06_ManageWindow
2. Amazon soyfasina gidelim. <https://www.amazon.com/>
3. Sayfanin konumunu ve boyutlarini yazdirin
4. Sayfayı simge durumuna getirin
5. simge durumunda 3 saniye bekleyip sayfayı maximize yapın
6. Sayfanin konumunu ve boyutlarini maximize durumunda yazdirin
7. Sayfayı fullscreen yapın
8. Sayfanin konumunu ve boyutlarini fullscreen durumunda yazdirin
9. Sayfayı kapatın



driver.manage().Method'lari

```
driver.manage().window().setPosition(new Point ( x:80 , y:0 ) );
```

Browser'i istenen koordinata tasir

```
driver.manage().window().setSize(new Dimension( 800 , 600 ));
```

Browser'i istenen olculere getirir.

```
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

Istenen sayfa acilincaya kadar maximum bekleme
suresini belirtir

```
driver.close(); sadece calisilan browser'i kapatir
```

```
driver.quit(); acik olan tum browser'lari kapatir
```



Selenium Navigation Methods

Class Work

1. Yeni bir Class olusturalim.C07_ManageWindowSet
2. Amazon soyfasina gidelim. <https://www.amazon.com/>
3. Sayfanin konumunu ve boyutlarini yazdirin
4. Sayfanin konumunu ve boyutunu istediginiz sekilde ayarlayın
5. Sayfanin sizin istediginiz konum ve boyuta geldigini test edin
8. Sayfayı kapatın



Homework

- 1.Yeni bir class olusturalim (**Homework**)
- 2.ChromeDriver kullanarak, facebook sayfasina gidin ve sayfa basliginin (title) “facebook” oldugunu dogrulayin (verify), degilse dogru basligi yazdirin.
- 3.Sayfa URL’inin “facebook” kelimesi icerdigini dogrulayin, icermiyorsa “actual” URL’i yazdirin.
- 4.<https://www.walmart.com/> sayfasina gidin.
5. Sayfa basliginin “Walmart.com” icerdigini dogrulayin.
6. Tekrar “facebook” sayfasina donun
7. Sayfayı yenileyin
- 8.Sayfayı tam sayfa (maximize) yapın
- 9.Browser’ı kapatın



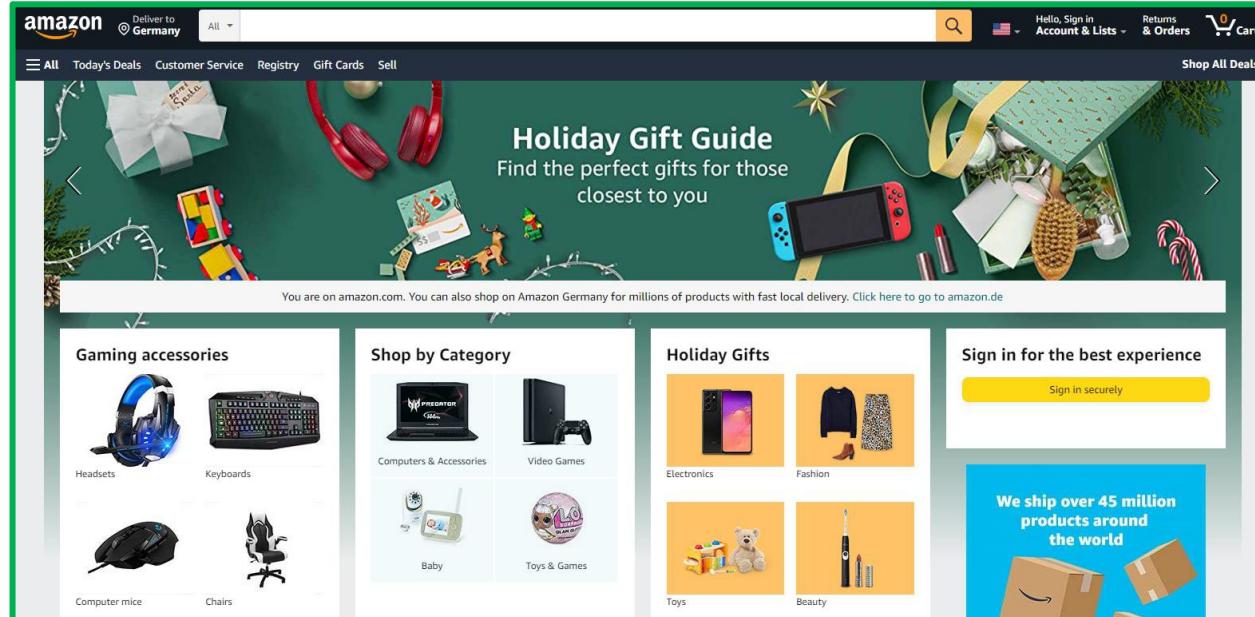
Tekrar Testi

1. Yeni bir class olusturun (**TekrarTesti**)
2. Youtube web sayfasına gidin ve sayfa başlığının “youtube” olup olmadığını doğrulayın (verify), eğer değilse doğru başlığı(Actual Title) konsolda yazdırın.
3. Sayfa URL'sinin “youtube” içerip içermediğini (contains) doğrulayın, içermiyorsa doğru URL'yi yazdırın.
4. Daha sonra Amazon sayfasına gidin <https://www.amazon.com/>
5. Youtube sayfasına geri donun
6. Sayfayı yenileyin
7. Amazon sayfasına donun
8. Sayfayı tamsayfa yapın
9. Ardından sayfa başlığının "Amazon" içerip içermediğini (contains) doğrulayın, Yoksa doğru başlığı(Actual Title) yazdırın.
10. Sayfa URL'sinin <https://www.amazon.com/> olup olmadığını doğrulayın, degilse doğru URL'yi yazdırın
11. Sayfayı kapatın



WEBELEMENTS

- Web sayfasında kullanılan etkilesimli olan veya olmayan herseye webelement denir



- **Button,**
 - **Search box(arama kutusu),**
 - **Text box(metin kutusu),**
 - **Headers(başlıklar),**
 - **Tables(tablolar) vb...**
- Farklı türde WebElement tag'ları(etiketleri) vardır.
- `<html>,<body>,<form>,<label>,<input>,<a> vb.`

- Otomasyon için unique(tek) web öğelerini(element) tanımlamak üzere HTML kodunu inceleyeceğiz(inspect).
- Web elementleri birlikte kullanıcı arayüzünde (UI) bir web sayfası oluştururlar.



WEBELEMENTS

```
▼<tbody>
  ▶<tr>...
  ▼<tr> == $0
    ▼<td>
      <input type="email" class="inputtext login_form_input_box" name="email" id="email" data-testid=
        "royal_email">
    </td>
    ▼<td>
      <input type="password" class="inputtext login_form_input_box" name="pass" id="pass" data-testid=
        "royal_pass">
    </td>
    ▼<td>
      ▶<label class="login_form_login_button uiButton uiButtonConfirm" id="loginbutton" for="u_0_b">...
    </label>
    </td>
  </tr>
  ▶<tr>...
</tbody>
```



Web Sayfalarını İnceleme (Inspect)

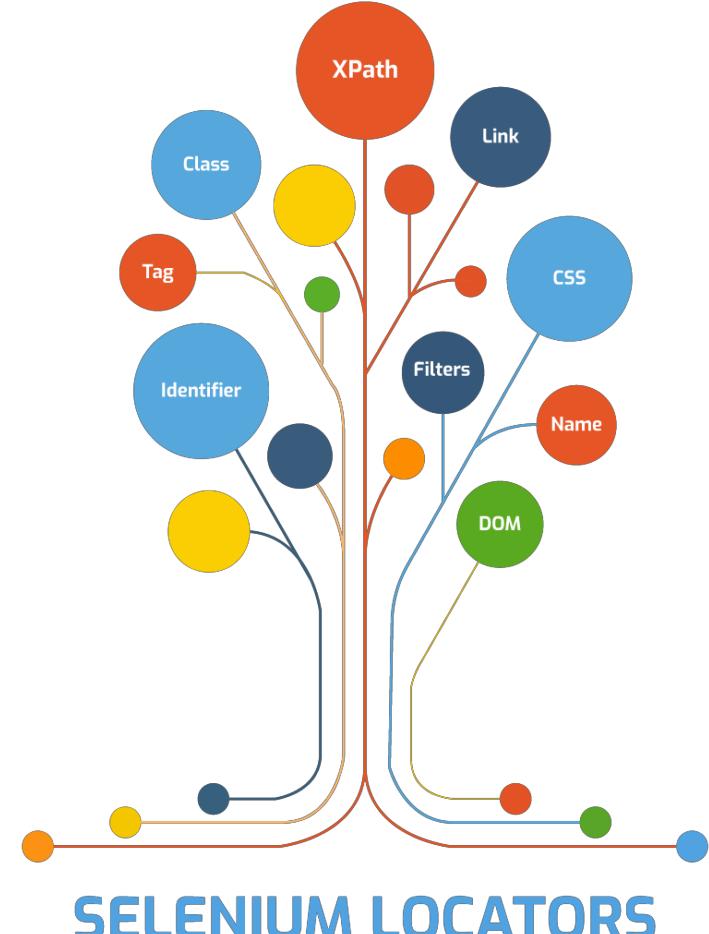
A screenshot of a web browser showing the developer tools' element inspector. The browser window displays the Amazon homepage. A context menu is open over the search bar, with the 'Inspect' option highlighted by a red oval. The browser's bottom bar shows tabs for Elements, Console, Sources, Network, Performance, Memory, and Application. The element inspector panel on the left shows the HTML structure of the page, with the search bar's input field selected.

```
<div class="nav-search">
  <div id="nav-bar-left"></div>
  <form accept-charset="utf-8" action="/s/ref=nb_sb_noss" class="nav-searchbar" method="GET" name="site-search" role="search">
    <div class="nav-left"></div>
    <div class="nav-right"></div>
  <div class="nav-fill">
    <div class="nav-search-field">
      <label id="nav-search-label" for="twotabsearchtextbox" class="aok-offscreen">
        Search
      </label>
      <input type="text" id="twotabsearchtextbox" value="" name="field-keywords" autocomplete="off" placeholder="Search" class="nav-input" dir="auto" tabindex="19" style="outline: none; border: 1px solid #d9d9d9; border-radius: 3px; width: 100%; height: 36px; font-size: 14px; font-family: inherit; padding: 0 10px; margin: 0; font-weight: normal; transition: border-color 0.15s ease-in-out, background-color 0.15s ease-in-out; position: relative; z-index: 1; ">
    </div>
  </div>
</div>
```



LOCATORS (YER BULUCU-KONUM BELIRLEYICI)

- ❖ Selenium LOCATORS, web sayfasındaki web öğelerini tanımlamak için kullanılır.
- ❖ Selenium'da; metin kutuları, onay kutuları, linkler, radyo butonları, liste kutuları ve **diğer tüm web öğeler** üzerinde eylemler gerçekleştirmek için LOCATORS'a ihtiyacımız vardır.
- ❖ Konum belirleyiciler bize nesneleri tanımlamada yardımcı olur.
- ❖ Web Elementlerine ulaşmak için tag veya bazı attribute'ler kullanılır, bunlarla ulaşamayan webelementleri için özel olarak tanımlanan **Xpath** ve css locator'lari kullanılır.





LOCATORS (YER BULUCU-KONUM BELIRLEYICI)

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords"
       autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute"
       dir="auto" tabindex="0" aria-label="Search">
```

Bir web elementini tanımlamak için 8 tane selenium locator vardır.

1. id
2. name
3. className
4. tagName
5. linkText
6. partialLinkText
7. **xpath** => xpath yazmanın birden fazla yolu vardır
8. **cssSelector** => css yazmanın birden fazla yolu vardır



LOCATORS BULMAK ICIN KULLANILAN METHOD'LAR

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords" autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute" dir="auto" tabindex="0" aria-label="Search">
```

1.) Tanimladigimiz web elementin yerini driver'in bulabilmesi icin findElement (Locator) method'unu kullaniriz.

```
driver.findElement(Locator);
```

2.) findElement (Locator) method'unun icine parameter olarak yazacagimiz Locator'in 8 locator'dan hangisi oldugunu belirtmek icin de By.LocatorTuru("LocateBilgisi"); kullanilir

```
driver.findElement(By.id("twotabsearchtextbox"));
```

- 3) Locate ettigimiz Web Elementini kullanabilmek icin bir variable'a atama yapariz

```
WebElement aramaKutusu = driver.findElement(By.id("twotabsearchtextbox));
```



LOCATORS

1) By.id();

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords"
       autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute"
       dir="auto" tabindex="0" aria-label="Search">
```

WebElement aramaKutusu = driver.findElement(By.id ("twotabsearchtextbox"));

- Web ögesini tanımlamanın en popüler yolu id kullanmaktadır.
- id en güvenli ve en hızlı locator seçeneği olarak kabul edilir ve her zaman birden çok locator arasında ilk öncelik olmalıdır.
- Eğer yanlış id locate edilirse; **NoSuchElementException** hatası oluşur.

*** **NoSuchElementException** gordugumuzde hata veren satirdaki locator gozden gecirilmelidir



LOCATORS

2)

By.name();

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords"
autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute"
dir="auto" tabindex="0" aria-label="Search">
```

WebElement passwordTextBox =driver.findElement(**By.name("field-keywords")**);

❖ Name ve value unique ise bu metodu da kullanabilirsiniz.



LOCATORS

3)

By.className():

```
<input class="form-control" placeholder="Password" data-test="password"
type="password" name="session[password]" id="session_password">
```

```
WebElement passwordTextBox =driver.findElement(By.className("form-control"));
```

- ❖ Class attribute'u olduğunda kullanılabilir.
- ❖ Class ve value unique ise, bu metodu da kullanabilirsiniz, ancak genelde class attribute aynı islevi yapan bir grup Web Elementi için kullanılır
- ❖ Class attribute'nun değeri boşluk içeriyorsa genelde By.className() ile yapılan locator'lar başarılı çalışmaz



LOCATORS

4) By.linkText();

```
<a class="nav-item nav-link" data-test="addresses" href="/addresses">Addresses</a>
```

```
WebElement passwordTextBox = driver.findElement(By.linkText("Addresses"));
```

- ❖ Bu yalnızca HTML bağlantılarını(link) tanımlamak için kullanılabilir.
- ❖ HTML link elementleri, bir web sayfasında bağlantı etiketi(tag) kısaltması olan <a> etiketi(tag) kullanılarak temsil edilir.
- ❖ Kullanıcı arayüzündeki(UI) hyperlinkleri kolayca tanıyalabilir ve sonra bu yöntemi kullanabilirsiniz
- ❖ Büyük / küçük harfe duyarlıdır (case sensitive) ve bağlantı(link) metniyle eşleşmelidir



LOCATORS

5) By.partialLinkText();

```
<a class="nav-item nav-link" data-test="addresses" href="/addresses">Addresses</a>
```

```
WebElement passwordTextBox = driver.findElement(By. partialLinkText("dresses"));
```

- ❖ `linkText ()` yöntemine benzer.
- ❖ Tek fark, tam metin vermek zorunda kalmamanızdır.
- ❖ Metnin yalnızca belirli bir bölümünü verebilirisiniz.
- ❖ Metnin tamamını verdiğinizde de kabul eder.



LOCATORS

6) By.tagName();

```
<input class="form-control" placeholder="Password" data-test="password"
type="password" name="session[password]" id="session_password">
```

```
WebElement passwordTextBox =driver.findElement(By.tagName("input"));
```

- ❖ Bu, diğer konum belirleyicilerden biraz farklıdır.
- ❖ <div>, <a>, <input>, ... gibi belirli bir etiketi ilettiğinizde, birden fazla aynı ad etiketine sahip olabileceğiniz için birden çok öğeyi döndürür.

Çoğunlukla öğelerin bir listesini almak için kullanılır. Bu nedenle findElements() yöntemiyle kullanılması önerilir.

Örneğin, kullandığımız bir sayfadaki tüm linkleri döndürmek için **By.tagName("a")** kullanılabilir



findElement() Method

WebElement

```
elementName=driver.findElement(By.LocatorStrategy("LocatorValue"));
```

- ❖ Driver'in bir elementi bulması için findElement() yöntemini kullanırız.
- ❖ Bu, tek bir web elementini döndürür. Aynı locator ile ulaşılabilen birden fazla web element varsa ilkini dondiurur
- ❖ Driver elementi bulamazsa, runtime exception verir :
NoSuchElementException.
- ❖ **NoSuchElementException**'ı gördüğünüzde, locatorı tekrar kontrol etmelisiniz.



findElements() Method

List<WebElement>

```
elementName=driver.findElements(By.LocatorStrategy("LocatorValue"));
```

- ❖ Locator degerine uygun Web elementlerinin listesini döndürür
- ❖ Locator stratejisiyle eşleşen web elementi yoksa boş bir liste döndürür.
- ❖ **NoSuchElementException** hatası vermez.
- ❖ Listedeki her Web elementi, 0'dan başlayan bir indeks alır.



findElements() ile findElement() arasindaki farklar

	findElement()	findElements()
websayfasinda birden fazla Web Element Locator ile uyusursa	Ilk elemani dondurur	Tum elemanlari ondurur
websayfasinda hicbir Web Element Locator ile uyusmazsa	NoSuchElementException fırlatir	Exception fırlatmaz, bos bir liste dondurur
Return Type	WebElement	List<WebElement>
Elemana erisim	Direk ulasilabilir	Liste'den index veya iterator ile ulasilabilir



WebElement Method'ları

Bir WebElement üzerinde eylemler gerçekleştirmek otomasyon tester'ları için çok önemlidir.

webElement.click(); Web Element'e click yapar

webElement.sendKeys("Metin"); Parametre olarak yazılan
metni
Web Elemente gönderir

webElement.submit(); Web element ile işlem yaparken
Enter tusuna basma görevi yapar

webElement.sendKeys("Metin" + Keys.ENTER); İstedigimiz metni yollayıp, sonra ENTER'a basar.



WebElement Method'lari

Class Work: Login Test

1. Bir class oluşturun: LoginTest
2. Main method oluşturun ve aşağıdaki görevi tamamlayın.
 - a. <http://a.testaddressbook.com> adresine gidiniz.
 - b. Sign in butonuna basin
 - c. email textbox,password textbox, and signin button elementlerini locate ediniz..
 - d. Kullanıcı adını ve şifreyi aşağıya girin ve oturum aç (sign in)buttonunu tıklayın:
 - i. Username : testtechproed@gmail.com
 - ii. Password : Test1234!
 - e. Expected user id nin testtechproed@gmail.com olduğunu doğrulayın(verify).
 - f. “Addresses” ve “Sign Out” textlerinin görüntülendiğini(displayed) doğrulayın(verify).
3. Sayfada kaç tane link olduğunu bulun.



WebElement Get Method'ları

webElement.getText(); Web Element üzerindeki yazıyı
getirir

webElement.getAttribute("Att.ismi"); İsmi girilen attribute'un
değerini getirir

webElement.getTagName(); Web elementin tag ismini
getirir

Not : Web element ile ilgili bu method'ların disinda size, istenen CSS ozelligi,
Location, Rect.geometrik ozellikler, DOM degeri vb.. Bir çok method vardır ama
Automation Test için kullanmıyoruz



WebElement Is Method'lari

webElement.isEnabled(); Web Element erisilebilir ise true
yoksa false doner

webElement.isDisplayed(); Web Element gorunur ise true
yoksa false doner

webElement.isSelected(); Web Element secili ise true
yoksa false doner

Not : Web element ile ilgili bu method'larin disinda size, istenen CSS ozelligi,
Location, Rect.geometrik ozellikler, DOM degeri vb.. Bir cok method vardir ama
Automation Test icin kullanmıyoruz



WebElement Method'ları

Tekrar Testi

1. Bir class oluşturun : AmazonSearchTest

2. Main method oluşturun ve aşağıdaki görevi tamamlayın.

- a. web sayfasına gidin. <https://www.amazon.com/>
- b. Search(ara) “city bike”
- c. Amazon'da görüntülenen ilgili sonuçların sayısını yazdırın
- d. Sonra karşınıza çıkan ilk sonucun resmine tıklayın.



HTML Parent-Child-Sibling Terimleri Nedir?

```
<div class="navFooterLine navFooterLinkLine navFooterDescLine" role="navigation" aria-label="More on Amazon.com">
  <table class="navFooterMoreOnAmazon" cellspacing="0">
    <tbody>
      <tr>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">
          <a href="https://advertising.amazon.com/?ref=footer_advtsing_amzn_com" class="nav_a">
            "Amazon Advertising"
            <br>
            <span class="navFooterDescText">...</span>
          </a>
        </td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
```

Tags: <> şeklinde görülen komutlara etiket (tag) adı verilir.

parent-child-sibling relationship(Ebeveyn-çocuk-kardeş ilişkisi) hakkında konuştuğumuzda, yalnızca tag adları önemlidir

<table>, <div> in çocuğudur(**child**) ve **<tbody>, <tr>** in ebeveyndir (**parent**)
<td> tagları ise siblings (kardeş)'dir



LOCATORS XPath

7) By.xpath();

Bir WebElement'i locate etmek icin kullanabilecegimiz en etkin method'dur.

```
WebElement passwordTextBox =driver.findElement(By.xpath("xpath"));
```

Method'un yazimi acisindan diger 6 yontem ile ayni olmakla beraber xpath'i digerlerinden ayiran cok onemli bir fark vardir

Diger 6 method HTML kod'una baglidir. Web element'in kodunda id yoksa By.id() method'unu, web element link degilse By.linkText() method'unu kullanamazsiniz.

Xpath ise dinamiktir. Her turlu web element icin mutlaka bir xpath yazilabilir

2 cesit Xpath yazilabilir

1.**Absolute** xpath (mutlak) 2.**Relative** xpath (bagil)





LOCATORS XPath

1. Absolute Xpath()

```
<div class="navFooterLine navFooterLinkLine navFooterDescLine" role="navigation" aria-label="More on Amazon.com">
  <table class="navFooterMoreOnAmazon" cellspacing="0">
    <tbody> <-- // div/ table/ tbody
      <tr>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">
          <a href="https://advertising.amazon.com/?ref=footer_advtsing_amzn_com" class="nav_a">
            "Amazon Advertising"
            <br>
            <span class="navFooterDescText">...</span> <-- // tbody / tr / td[3] // span
          </a>
        </td> <-- // tbody / tr / td[3] / a / span
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
```

Absolute xpath yazmak icin en basa // sonraki her adimda / yazarak hedef web element'e kadar tum tag'lar yazilir.

Eger ayni path'e sahip birden fazla element varsa index kullanilabilir. [2] gibi

Eger bir parent'in grand child'lari icinde unique bir tag varsa parent // grand child yazilabilir



LOCATORS XPath

2. Relative Xpath()

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords" autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute" dir="auto" tabindex="0" aria-label="Search">
```

Bir web element'te temel olarak 3 bileşen vardır.
1) tag name 2) attribute input

3) attribute value type, id , value, name, autocomplete, placeholder, class

type → 'text', id→ 'twotabsearchtextbox' , value → ''

Bu 3 bileşeni birlikte kullanarak her bir webelement için unique sonuc veren bir çok xpath yazılabilir

Relative xpath yazmak için bu 3 bileşen aşağıdaki gibi bir araya getirilir, unique sonuc veren her relative xpath kullanılabilir.

//tagName[@attributelsmi='attributeValue']



LOCATORS XPath

2. Relative Xpath()

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords" autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute" dir="auto" tabindex="0" aria-label="Search">
```

- ❖ Genelde 3 bilesen de kullanılır, Ancak bazen daha az bilesen yazmak da yeterli olabilir.

Sadece tag name kullanarak xpath yazmak için

```
driver.findElement(By.xpath("//input" ));
```

Tag name farketmeksizin attribute ismi ve attribute value kullanarak xpath yazmak için

```
driver.findElement(By.xpath("// * [@type='text']" ));
```

Attribute name farketmeksizin tag name ve attribute value kullanarak xpath yazmak için

```
driver.findElement(By.xpath("//input[@ *=‘text’]" ));
```

Attribute value farketmeksizin tag name ve attribute ismi kullanarak xpath yazmak için

```
driver.findElement(By.xpath("//input[@type]" ));
```



LOCATORS XPath

Class Work: Add Remove Element

- 1- https://the-internet.herokuapp.com/add_remove_elements/ adresine gidin
- 2- Add Element butonuna basin
3. Delete butonu'nun gorunur oldugunu test edin
4. Delete tusuna basin
5. “Add/Remove Elements” yazisinin gorunur oldugunu test edin



LOCATORS XPath

2. Relative Xpath()

- ❖ Bazen de attribute'e bagli olmadan sadece web element icinde bulunan text kullanilabilir.

Exact Text(Belirli bir text) ile element bulma:

```
driver.findElement(By.xpath("//tagname[.= 'text name'] "));  
driver.findElement(By.xpath("//*[.= 'text name'] " )); driver.findElement(By.xpath("//*[text( )='exact text with extra space and all'] " ));
```

Belirli bir metni içeren bir öğeyi bulmak için şunları kullanabiliriz:

```
driver.findElement(By.xpath("//*[contains(text(),'piece of text')] " ));
```

- ❖ Tek attribute ile unique bir sonuca ulasamazsak birden fazla attribute yazabiliriz

```
driver.findElement(By.xpath("//div[@id='logo' or class='flex-col logo' ] "));  
driver.findElement(By.xpath("//div[@id='logo' and class='flex-col logo' ] "));
```



LOCATORS XPath

Class Work: Add Remove Element

Asagidaki testi text'leri kullanarak locate edin

1. https://the-internet.herokuapp.com/add_remove_elements/ adresine gidin
2. Add Element butonuna basin
3. Delete butonu'nun gorunur oldugunu test edin
4. Delete tusuna basin
5. “Add/Remove Elements” yazisinin gorunur oldugunu test edin



LOCATORS CssSelector

8- By.cssSelector()

Method

```
<input class="form-control" placeholder="Password" data-test="password"
type="password" name="session[password]" id="session_password">
```

Css selector xpath'e benzer. Üç ana tip kullanılır

1) css = tagName[attribute name= 'value'];

```
driver.findElement(By.cssSelector("input[name='session[password]']"));
```

2.) css="tagName#idValue" veya sadece css="#idValue" =>yalnızca id value ile çalışır

```
driver.findElement(By.cssSelector("input#session_password"));
```

3.) css="tagName.className" veya sadece css=".className"=>yalnızca class value ile çalışır

```
driver.findElement(By.cssSelector(".form-control"));
```



LOCATORS CssSelector

Home Work: Log in Test Using Css

- 1.) Bir class oluşturun : Locators_css
- 2.) Main method oluşturun ve aşağıdaki görevi tamamlayın.
 - a. Verilen web sayfasına gidin. [http://a.testaddressbook.com/
sign_in](http://a.testaddressbook.com/sign_in)
 - b. Locate email textbox
 - c. Locate password textbox ve
 - d. Locate signin button
 - e. Asagidaki kullanıcı adını ve şifreyi girin ve sign in düğmesini tıklayın
 - i. Username : testtechproed@gmail.com
 - ii. Password : Test1234!

NOT: cssSelector kullanarak elementleri locate ediniz.



LOCATORS XPath

Tekrar Testi

1-C01_TekrarTesti isimli bir class olusturun

2- <https://www.amazon.com/> adresine gidin

3- Browseri tam sayfa yapın

4.- Sayfayı “refresh” yapın

5. Sayfa basliginin “Spend less” ifadesi icerdigini test edin

6. Gift Cards sekmesine basin

7. Birthday butonuna basin

8. Best Seller bolumunden ilk urunu tiklayin

9. Gift card details'den 25 \$'i secin

10-Urun ucretinin 25\$ oldugunu test edin

10-Sayfayı kapatın



LOCATORS Relative Locators

Relative Locators nedir ?

- ❖ Selenium 4 ile gelen yeniliklerden biri de bagil locator'lardir.
- ❖ Bir web elementi direk locate edemedigimiz durumlarda gunluk hayatimizda kullandigimiz sekilde o web elementi etrafindaki web elementlerin referansi ile tarif edebiliriz.
- ❖ Ornegin yandaki resimde Berlin icin bir cok relative locator tanimlayabiliriz.
 - Boston'in saginda , Sailor'in ustunde
 - NYC'nin altinda, Bay Area'nin solunda
 - Boston yakinlarinda Bay Areanin solunda ve Toronto'nun saginda vb..



<https://www.diemol.com/selenium-4-demo/relative-locators-demo.html>



LOCATORS Relative Locators

Class Work: Relative Locators

```
driver.get("https://www.diemol.com/selenium-4-demo/relative-locators-demo.html#");

WebElement boston=driver.findElement(By.id("boston"));
WebElement sailor = driver.findElement(By.id("sailor"));

WebElement berlin = driver.findElement(with(By.tagName("li")).above(sailor).toRightOf(boston));

WebElement mountie=driver.findElement(with(By.className("ui-li-has-thumb")).below(boston));
```





LOCATORS Relative Locators

Class Work: Relative Locators

1.) <https://www.diemol.com/selenium-4-demo/relative-locators-demo.html> adresine gidin

2.) Berlin'i 3 farkli relative locator ile locate

edin 3) Relative locator'larin doaru calistilagini test edin





Maven™

Bir projeyi geliştirmek kadar onu anlasılır kılmak, tekrar eden kodlardan kaçınmak, updateleri tek merkezden kolayca yapabilmek ve projenin surdurulebilirliğini sağlamak (maintenance) da önemlidir.

- ❖ Apache Maven bir yazılım proje yönetimi ve anlama aracıdır.
- ❖ Maven, en iyi uygulamaların geliştirilmesi için güncel ilkeleri bir araya getirmeyi ve bir projeye bu yönde rehberlik etmeyi kolaylaştırmayı amaçlamaktadır.
- ❖ Maven geliştiricileri birçok ayrıntıdan korur.
- ❖ Proje nesne modeli (POM Project Object Model) konseptine dayalıdır.

Apache
Maven™

[https://
maven.apache.org/](https://maven.apache.org/)

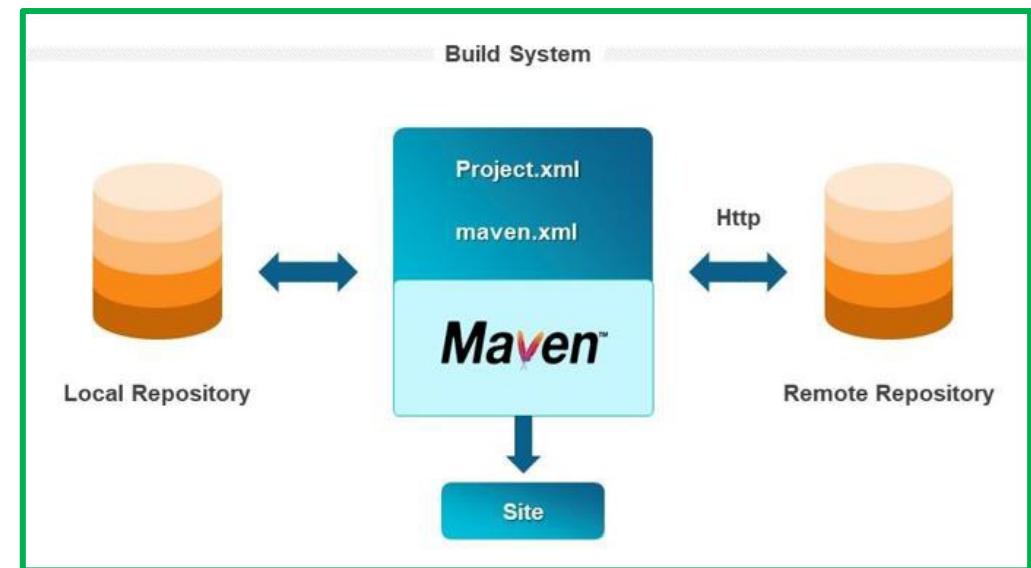


Maven™

- ❖ Maven bir Java derleme aracıdır (build tool). Maven proje otomasyon ve yönetim aracıdır (automation and management tool).

- ❖ Maven, konfigürasyon için pom.xml dosyasını kullanır. Bu dosya projenin insası, raporlaması ve dokümantasyonu için gerekli bütün bilgileri içerir (dependencies, plugins)

- ❖ Bir Maven projesine aşina olduğunuzda, tüm Maven projelerinin nasıl inşa edildiğini bilirsiniz. Bu, birçok projede gezinirken zaman kazandırır.





Maven™

Neden Maven?

- ❖ Tekrarlanabilir derlemeler / yürütütmeleri yapmak kolaylaştırır.(Repeatable builds/ executions.)
- ❖ Proje yönetimini kolaylaştırır.
- ❖ Maven, dependencylerle ilgili mevcut jar dosyalarını otomatik olarak indirir.
- ❖ Birden fazla IDE (intelliJ, eclipse, vb.) ve araçlarla(tools) çalışır.
- ❖ Open source
- ❖ Geniş kullanıcı tabanı



POM XML

✓ Pom.xml dosyası nedir?

Maven projesindeki en onemli dosyadır. Dependencies'i POM dosyasına ekler ve POM dosyasından yönetiriz.

✓ Maven ve pom arasındaki fark nedir?

Maven bir araç (tool), pom bir xml dosyasıdır. Pom.xml maven'in bir parçasıdır ve pom dependencies'i yönetir.

✓ Projenizde pom'ı nasıl kullanız? Neden kullanız?

Dependencies'i yönetmek için . Tüm projeyi yönetmemize yardımcı olur. Pom ayrıca artifact id, group id, ve version gibi proje bilgilerine de sahiptir.

✓ Dependency nedir? Nasıl kullanıyorsunuz? Neden kullanıyorsunuz?

Sürücüler (drivers) kurmak, sürücüler oluşturmak için dependencies gereklidir. Gerekli araçları içe aktarır(import).

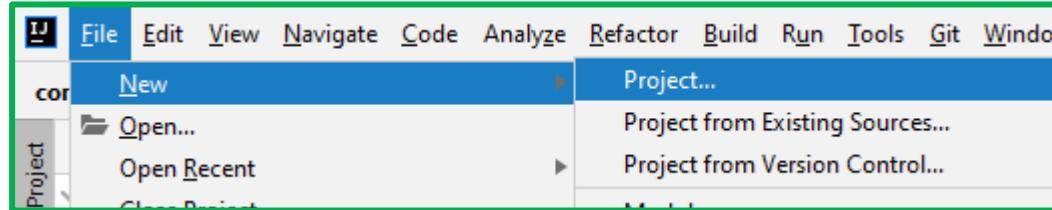
✓ Pom dosyanızı ve dependencies'i nasıl güncellersiniz ?

Mvnrepository.com adresinden gerekli dosyalari bulur ve pom dosyamiza ekleriz



Maven™

Maven PROJESINI OLUSTURMA



1. Create Project: File -> New -> Project

2. Select Maven -> click next
Java versiyonunun bilgisayarinizdaki version ile ayni oldugunu control edin

3.Name: com.Batch81Maven -> finish -> click on (EnableAutoImport)

4.Package olusturun, name : day04 -> Right click on java create the package
5.Class olusturun, name : FirstMavenClass



Maven™

POM XML DOSYASINA DEPENDENCIES EKLEME

- 1) Mvnrepository.com adresine gidelim
- 2) WebDriverManager aratalim (En guncel ve en cok kullanılan optimum versiyonu alalim)
- 3) Asagida Maven yazan bolumdeki kodlari kopyalayalim 4) pom XML dosyasina gidelim
- 5.) <properties> satirinin altina <dependencies> </dependencies> tag'i olusturalim
- 6.) Kopyaladigimiz kodlari <dependencies> bolumune yapistiralim 7) Ayni islemleri Selenium Java dependency icin de yapalim
8.) Pom XML dosyasinin sag tarafinda **Maven** yazan bolumu tiklayalim
9.) Yenile butonuna tiklayip asagida Dependencies bolumunun olustugunu ve altinda ekledigimiz kutuphanelerin gorundugunden emin olun



Maven™

POM XML DOSYASINA DEPENDENCIES EKLEME

The screenshot shows the Maven Project Structure on the right and the pom.xml file content on the left.

Maven Project Structure:

- com.mavenSe4
- Lifecycle
- Plugins
- Dependencies
 - io.github.bonigarcia.webdrivermanager:5.0.3
 - org.seleniumhq.selenium:selenium-java:4.1.0

pom.xml Content:

```
<groupId>org.example</groupId>
<artifactId>com.mavenSe4</artifactId>
<version>1.0-SNAPSHOT</version>

<properties>
    <maven.compiler.source>11</maven.compiler.source>
    <maven.compiler.target>11</maven.compiler.target>
</properties>

<dependencies>
    <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
    <dependency>
        <groupId>io.github.bonigarcia</groupId>
        <artifactId>webdrivermanager</artifactId>
        <version>5.0.3</version>
    </dependency>

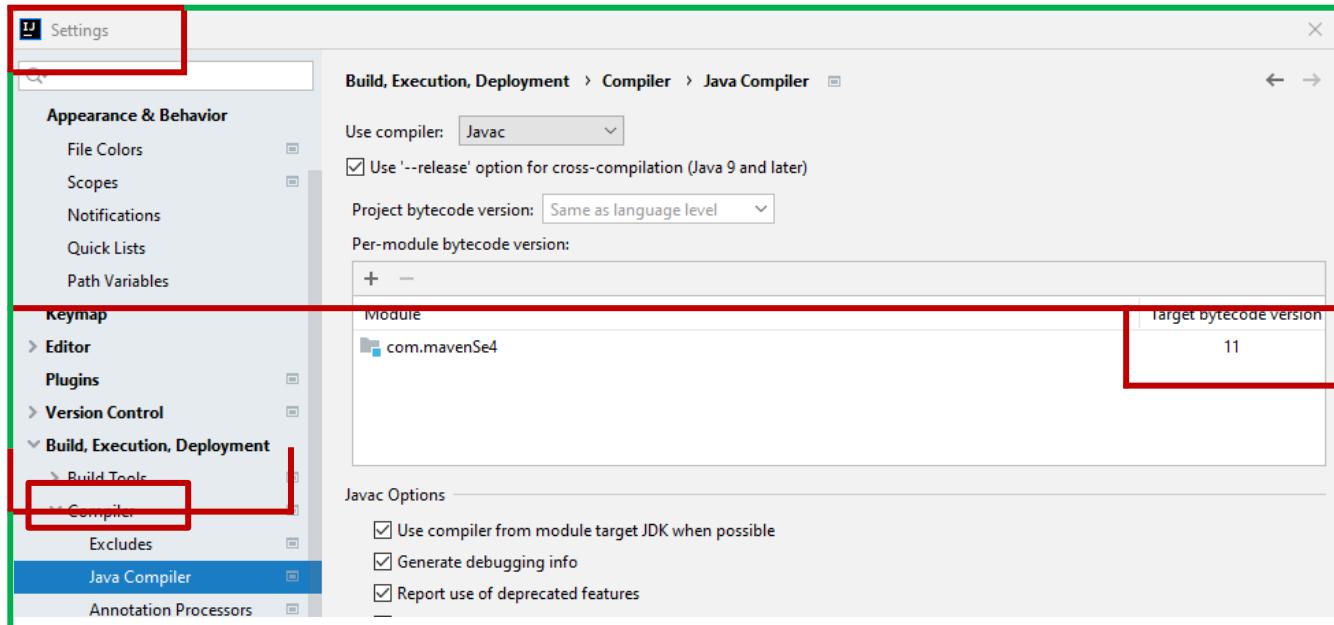
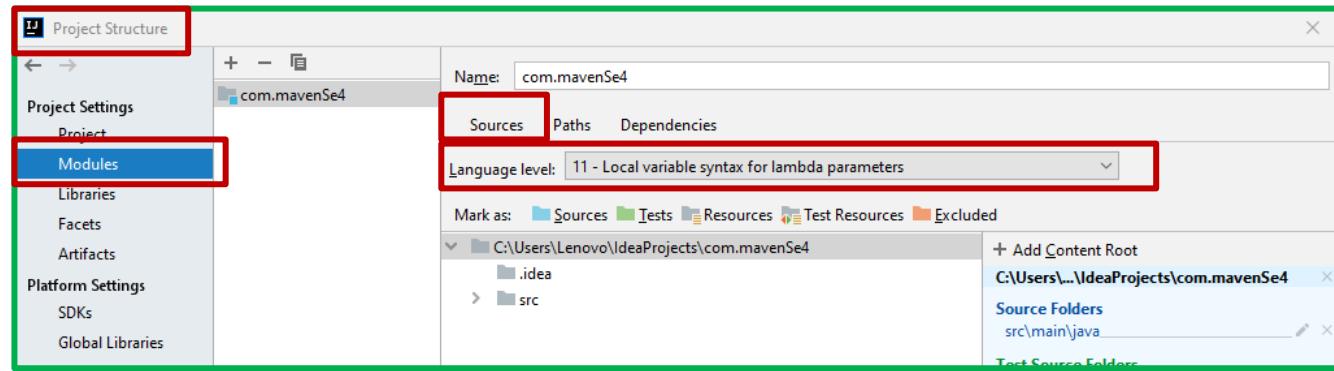
    <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>4.1.0</version>
    </dependency>
</dependencies>
</project>
```

The dependencies section is highlighted with a red box. The Maven icon in the toolbar and the Maven tab in the project tree are also highlighted with red boxes.



Maven™

CLASS AYARLARI



File
Project
Structure
Modules
Sources

Language level : min **8** yapalim,
bilgisayarinizda kurulu java 8 veya ustу
ise **java versiyonu aynи olmali**

File
Settings
Build,Execution,Deployment
Compiler
Java Compiler

Target bytecode version : min **8** yapalim,
bilgisayarinizda kurulu java 8 veya ustу
ise **java versiyonu aynи olmali**



Maven™

CLASS WebDriver AYARLARI

```
WebDriverManager.chromedriver( ).setup  
( ); WebDriver driver = new  
ChromeDriver();
```

Class Work Amazon Search Test

- 1. <https://www.amazon.com/> sayfasina gidelim**
- 2. arama kutusunu locate edelim**
- 3. “Samsung headphones” ile arama yapalim**
- 4. Bulunan sonuc sayisini yazdiralim**
- 5. Ilk urunu tiklayalim**
- 6. Sayfadaki tum basliklari yazdiralim**



Soru 1

1. <http://zero.webappsecurity.com> sayfasina gidin
2. Signin buttonuna tiklayin
3. Login alanine “username” yazdirin
4. Password alanine “password” yazdirin
5. Sign in buttonuna tiklayin
6. Pay Bills sayfasina gidin
7. amount kismina yatirmak istediginiz herhangi bir miktarı yazin
8. tarih kismina “2020-09-10” yazdirin
9. Pay buttonuna tiklayin
10. “The payment was successfully submitted.” mesajinin ciktigini control edin



Tekrar Testi

- 1-C01_TekrarTesti isimli bir class olusturun 2- <https://www.google.com/> adresine gidin 3- cookies uyarisini kabul ederek kapatın

4. Sayfa basliginin “Google” ifadesi icerdigini test edin
5. Arama cubuguna “Nutella” yazip aratin
6. Bulunan sonuc sayisini yazdirin
7. sonuc sayisinin 10 milyon’dan fazla oldugunu test edin
8. Sayfayı kapatın



Soru 3

1. "<https://www.saucedemo.com>" Adresine gidin
2. Username kutusuna "standard_user" yazdirin
3. Password kutusuna "secret_sauce" yazdirin
4. Login tusuna basin
5. Ilk urunun ismini kaydedin ve bu urunun sayfasina gidin
6. Add to Cart butonuna basin
7. Alisveris sepetine tiklayin
8. Sectiginiz urunun basarili olarak sepete eklendigini control edin
9. Sayfayı kapatın



JUnit

- ❖ Java ile en temel framework JUnit ile oluşturulabilir.
- ❖ Developerlar da unit testleri calistirmak icin kullanirlar.
- ❖ Biz testlerimizi yapmak icin JUnit'in ileri sürümü olduğundan TestNG framework oluşturup kullanacagiz.
- ❖ Junit maven projesi üzerinde calistigindan <https://mvnrepository.com/> sitesinden dependency'leri projemize ekleriz.
- ❖ Test'lerimizi yaparken main method, if-else gibi java kod bloklarini kullanmak yerine Junit annotationlarini ve method'larine kullanabiliriz.
- ❖ Son framework olarak kuracagimiz Cucumber framework'da da Junit kullanacagiz





Annotations

- ❖ Selenium'da kodlarımızı yazarken “@” işaretini ile başlayan notasyonlar kullanırız.
- ❖ Java Annotationlar ile derleyiciye (Compiler) talimatlar verebiliriz.
- ❖ Annotation, bir veri hakkında bilgi barındıran veriyi sağlayan basit bir yapıdır. Bu sağladığı bilgiye de “metadata” denir.
- ❖ Notasyonlar(Annotation) genellikle Java'da konfigürasyon amacıyla kullanılır. Kullanıldığı bileşene ek özellikle katar. Bu bileşenler sınıf, metod, değişkenler, paket ya da parametreler olabilir. Bunların hepsinde notasyonları kullanabiliriz.

En çok kullanılan Junit annotation'ları

`@Test`

`@BeforeClass @AfterClass`

`@Before , @After`

`@Ignore`



Annotations

@Test ve @Ignore

- ❖ Junit ile Main Method kullanma donemini bitiriyoruz.
- ❖ Junit Framework kullandigimizda yazdigimiz test metodunun calismasi icin başına @Test notasyonu eklememiz yeterlidir.
- ❖ @Test notasyonu eklemedigimiz metot test sirasinda calisitirilmaz. Ancak cagrilrsa calisir.
- ❖ Yazdigimiz bazı test metotları henüz tamamlanmamış veya değişikliklere uğrayabileceginden dolayı test sirasinda calistirilmasini istemiyorsak @Ignore notasyonu eklememiz yeterlidir.
- ❖ @Ignore notasyonun tanimli olduğu metotlar test sirasinda calistirilmayacaktir. Ayrıca istenilirse @Ignore("açıklama") şeklinde yazılarak metodun neden test edilmesini istemedigimizde yazabiliriz.



Annotations

@Before ve @After

- ❖ Before notasyonu, her test method'undan önce çalışır. Örneğin bir sayfa ile test yapıyorsak ve her testten önce o sayfaya gitmemiz gerekiyorsa @before kullanabiliriz.
@before notasyanunun kullanıldığı method'a genelde setup() ismi verilir.

- ❖ After notasyonu, her test method'undan sonra çalışmaktadır. Örneğin test sırasında kullandığımız sayfanın kapatılması gibi.
@after notasyanunun kullanıldığı metoda genelde teardown() ismi verilir

Not : Test method'u ve Test farklı yapılardır.

Test dedigimizde tek bir method veya içinde bircok class ve package barindiran bir yapı olabilir. Regression test, smoke test vb..

Test method'u ise @Test notasyonu kullanılarak olusturulan ve bagimsiz olarak calistirabilecegimiz en kucuk test yapısıdır



Annotations

@BeforeClass ve @AfterClass

- ❖ BeforeClass notasyonu, bir class'daki tüm testlerden önce yapılması gereken bir işlem işlem varsa kullanılır (precondition).
Örneğin test metodlarımız çalışmadan driver olusturup tüm methodlarda kullanabilirim.

- ❖ AfterClass notasyonu da, bir class'daki tüm testler tamamlandıktan sonra yapılması gereken işlemlerde kullanılır.
Örneğin actigimiz sayfayı kapatmak veya elde ettigimiz test sonuclarını raporlamak gibi.

- ❖ @BeforeClass ve @AfterClass notasyonları test sürecinde bir kere çalışırken, @Before ve @After notasyonları her test method'unun başında ve sonunda çalışmaktadır.



JUnit

Annotations

@BeforeClass - @AfterClass ve @Before - @After notasyonlarinin karsilastirilmasi

1.) Tum Class calistirilirsa

```
Before class method'u calisti → @BeforeClass  
Before method'u calisti  
Test 1 calisti } @Before - @After  
After method'u calisti }  
Before method'u calisti } @Before - @After  
Test 2 calisti }  
After method'u calisti } @Before - @After  
Before method'u calisti }  
Test 3 calisti }  
After method'u calisti }  
After class method'u calisti → @AfterClass
```

2.) Tek bir test method'u calistirilirsa

```
Before class method'u calisti → @BeforeClass  
Before method'u calisti } @Before - @After  
Test 3 calisti }  
After method'u calisti }  
After class method'u calisti → @AfterClass
```

```
public class C03 {  
    @BeforeClass  
    public static void beforeClassM(){  
        System.out.println("Before class method'u calisti");  
    }  
    @Before  
    public void beforeM() {  
        System.out.println("Before method'u calisti");  
    }  
    @Test  
    public void test1(){  
        System.out.println("Test 1 calisti");  
    }  
    @Test  
    public void test2(){  
        System.out.println("Test 2 calisti");  
    }  
    @Test  
    public void test3(){  
        System.out.println("Test 3 calisti");  
    }  
    @AfterClass  
    public static void afterClassM(){  
        System.out.println("After class method'u calisti");  
    }  
    @After  
    public void afterM() {  
        System.out.println("After method'u calisti");  
    }  
}
```



- 1.Bir class oluşturun : CheckBoxTest**
- 2. Gerekli yapıyi olusturun ve aşağıdaki görevi tamamlayın.**
 - a. Verilen web sayfasına gidin.**
<https://the-internet.herokuapp.com/checkboxes>
 - b. Checkbox1 ve checkbox2 elementlerini locate edin.**
 - c. Checkbox1 seçili değilse onay kutusunu tıklayın**
 - d. Checkbox2 seçili değilse onay kutusunu tıklayın**



RadioButton

1. Bir class oluşturun : RadioButtonTest

2. Gerekli yapıyi olusturun ve aşağıdaki görevi tamamlayın.

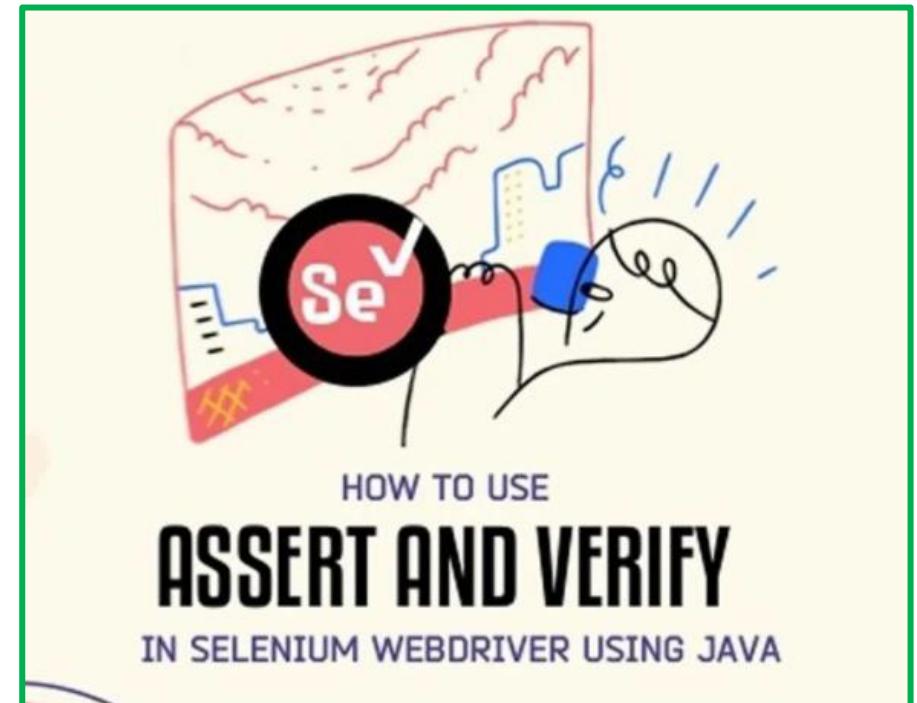
- <https://www.facebook.com> adresine gidin
- Cookies'i kabul edin
- “Create an Account” button'una basin
- “radio buttons” elementlerini locate edin
- Secili degilse cinsiyet butonundan size uygun olanı secin



JUnit

Assertions: (Verification)

- ❖ Test icin uygun bir sekilde hazırlanmış bir framework'da expected değerlerin actual değerlere eşit olup olmadığını if-else statement ile bakılmaz.
- ❖ JUnit framework'ünde test senaryomuzu çalıştırıldıktan sonra beklediğimiz sonuçların olup olmadığını tespit etmek için assert class'ından method'lar kullanırız.
- ❖ Junit'de, belirli koşulları test etmek için isimleri assert ile başlayan statik method'lari kullanırız.
- ❖ Bu method'lar ile excepted ve actual değerleri karşılaştırarak testimizi sonuçlandırırız.





Assertions

- ❖ Assert method'unu secmeden once kiyaslamada bekledigimiz sonucun true veya false olmasina karar vermeliyiz.

```
String exceptedName= "Ali Can";
String actualName= "ALI CAN";

exceptedName.equals(actualName); → fals
exceptedName.equalsIgnoreCase(actualName); → true
```

// Emeklilik yasi 65

int yas1=63;

int yas2=68;

emekliOlabilirMi(yas1); → fals

emekliOlabilirMi(yas2); → true

- ❖ ONEMLI OLAN kiyaslama sonucunun true veya false olması değil, bekledigimiz sonucun olup olmamasidir. Assertion başarısız olursa AssertionException ile hata mesajı verilir.

Assert.assertEquals(actualName,expectedName) → FAIL
 fals
 D

Assert.assertTrue(yas2>65) → PASSED
 true
 D

Assert.assertTrue(yas2<65) → FAIL
 fals
 D

Assert.assertFalse(yas1>65) → PASSED
 fals
 D

Assert.assertFalse(yas1<65) → FAIL
 true
 D



Assertions

A=2 B=3 C=4 D=2
0 0 0 0

Assert.assertEquals(A,B) → FAIL
 |
 false
 e

Assert.assertEquals(A,D) → PASS
 |
 true

Assert.assertTrue(C>D) → PASS
 |
 true

Assert.assertTrue(C>B) → PASS
 |
 true

Assert.assertFalse(B>D) → FAIL
 |
 true

Assert.assertFalse(B>A) → FAIL
 |
 true



Assertions

1) Bir class oluşturun: **YoutubeAssertions**

2) <https://www.youtube.com> adresine gidin

3) Aşağıdaki adları kullanarak 3 test metodu oluşturun ve gerekli testleri yapın

=> Sayfa başlığının “YouTube” olduğunu test edin

=> YouTube resminin görüntüldüğünü (`isDisplayed()`) test edin

○ `Search Box` 'in erişilebilir olduğunu test edin (`isEnabled()`)

○ `wrongTitleTest` => Sayfa basliginin “youtube” olmadığını doğrulayın



Assertions

1. Bir Class olusturalim YanlisEmailTesti
2. <http://automationpractice.com/index.php> sayfasina gidelim
3. Sign in butonuna basalim
4. Email kutusuna @isareti olmayan bir mail yazip enter'a bastigimizda “Invalid email address” uyarisi ciktigini test edelim



ASSERT OZET

Assert, bir test senaryosunun PASS veya FAILED durumunu belirlemede kullanılan yararlı bir yöntemdir. Secilen metod ve yazılan **boolean koşul'a** göre test sonucu belirlenir.

En çok kullandığımız 3 Assert metodu;

1) Assert.assertTrue(**koşul**)

Yazılan koşul'un sonucu True ise test PASS, yoksa test FAILED olur

Assert.assertTrue(20 > 15) → Test PASSED



Assert.assertTrue(10 > 30) → Test FAILED



False



ASSERT OZET

2) Assert.assertFalse(**koşul**)

Yazilan koşul'un sonucu False ise test PASS, yoksa test FAILED olur

Assert.assertFalse(40 > 50) → Test PASSED
False

Assert. assertFalse(30 > 20) → Test FAILED
True

3) Assert.assertEquals(**expected, actual**)

Yazilan expected ile actual esit ise test PASS, yoksa test FAILED olur

Assert.assertEquals("Ali" , "Ali") → Test PASSED
True

Assert. assertEquals(30 , 20) → Test FAILED
False



JUnit

Handle Dropdown

- **Adım1:** Dropdown menüyü herhangi bir locator ile locate edin.

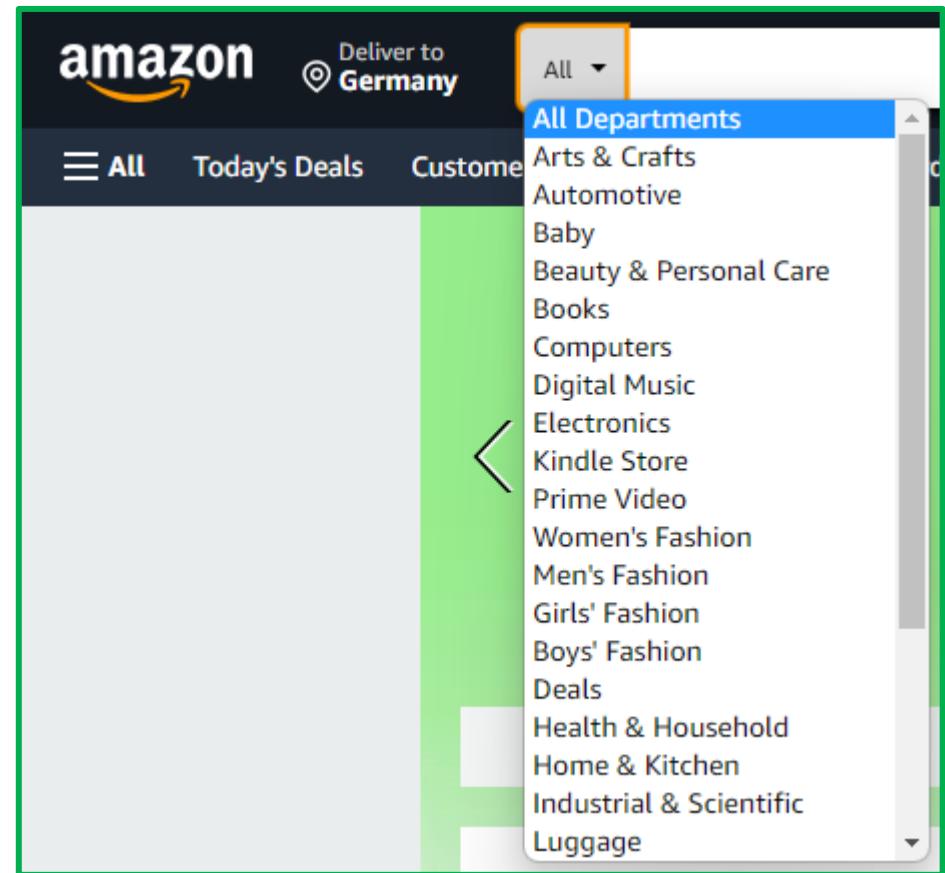
```
WebElement ddm=driver.findElement(By.id("value of id"));
```

- **Adım 2:** Yeni bir “Select” objesi olusturun ve daha önce locate ettigimiz WebElement'i parametre olarak yeni objeye ekleyin

```
Select options=new Select(ddm);
```

- **Adım 3:** Select class'indan kullanabileceginiz 3 yontemden biriyle dropdown menusundeki elemanlardan istediginizi secin

```
options.selectByIndex(1);
```





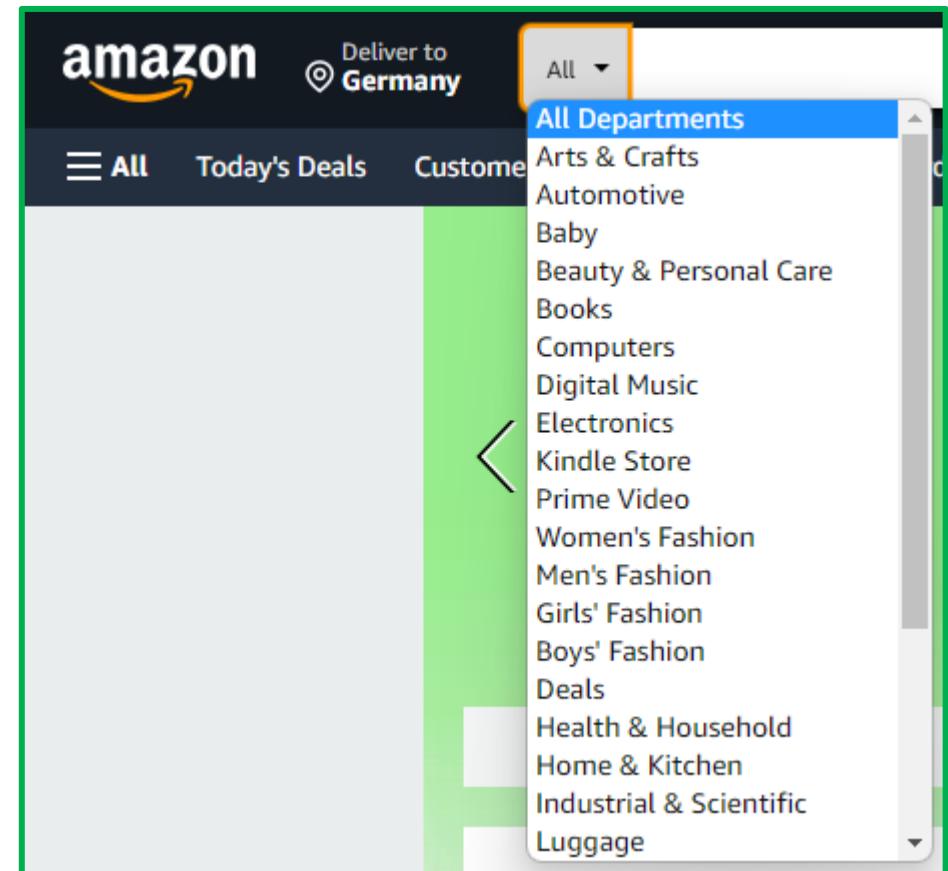
JUnit

Handle Dropdown

Dropdown menusundeki elementleri Select Class'ından kullanacagimiz yontemlerle 3 sekilde secebiliriz

1. Index kullanarak **selectByIndex();**
2. Deger kullanarak **selectByValue();**
3. Gorunen degerini kullanarak **selectByVisibleText();**

Istenirse **getOptions();** methodu kullanilarak DropDown'daki tum elementler webelementlerden olusan bir listeye konabilir. **List<WebElement>**





Handle 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 öğe varsa konsolda True , degilse False yazdırın.



Handle Dropdown

- Bir class oluşturun: C3_DropDownAmazon
- <https://www.amazon.com/> adresine gidin.
 - **Test 1**

Arama kutusunun yanindaki kategori menusundeki kategori sayisinin 45 oldugunu test edin

-Test 2

- 1.Kategori menusunden Books secenegini secin
2. Arama kutusuna Java yazin ve aratin
3. Bulunan sonuc sayisini yazdirin
4. Sonucun Java kelimesini icerdigini test edin



Handle Dropdown

1. <http://zero.webappsecurity.com/> Adresine gidin
2. Sign in butonuna basin
3. Login kutusuna “username” yazın
4. Password kutusuna “password.” yazın
5. Sign in tusuna basin
6. Pay Bills sayfasına gidin
7. “Purchase Foreign Currency” tusuna basin
8. “Currency” drop down menusünden Eurozone’u secin
9. “amount” kutusuna bir sayı girin
10. “US Dollars” in secilmediğini test edin
11. “Selected currency” butonunu secin
12. “Calculate Costs” butonuna basin sonra “purchase” butonuna basin
13. “Foreign currency cash was successfully purchased.” yazısının çıktığını kontrol edin.



En sonki dersimizden neler hatırlıyoruz?

Dropdown?

-Acılır menu. Birden fazla seçeneği koyabileceğimiz liste.

***Dropdown nasıl automate edilir? How to handle dropdown?

1. Dropdown elementi locate edilir. 8 locator'dan biri kullanılabilir.

2. Select classı kullanılarak bir Select objesi oluşturulur. Select select = new Select(dropdownElement);

3. Select objectini kullanarak uygun methodlarla işlem yapılır

- selectByIndex(int); index 0 dan başlar

- selectByValue("String");

- selectByVisibleText("String");

***getOptions(); tüm dropdown seçenekleri List<WebElement> olarak return eder

***getFirstSelectedOption(); seçili olana ilk seçenek(default) return eder.



JUnit

TestBase Class

- TestBase, testlerden önce ve sonra tekrar tekrar kullandığımız kodları içermektedir.
- İçerisindeki metodları kullanabilmemiz için extends yapıyoruz. Bu sayede test class'ımızda sadece test case'ler bulunmaktadır.
- Utilities package'da TestBase'i oluşturuyoruz.
 - setUp method
 - reports (Raporlar)
 - tearDown method
- TestBase class'i abstract yapabiliriz (Olmasada olabilir) , extends yaparak kullanabiliriz. Ve bu class'da object create edemeyiz.
- olusturdugumuz driver'in farklı package'lardan kullanılmak için PROTECTED yaparız

```
import java.util.concurrent.TimeUnit;

public abstract class TestBase {
    protected WebDriver driver;
    @BeforeMethod
    public void setUp() {
        WebDriverManager.chromedriver().setup();
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().pageLoadTimeout(10, TimeUnit.SECONDS);
        driver.manage().implicitlyWait(10, TimeUnit.SECONDS);
    }

    @AfterMethod
    public void tearDown(){
        driver.quit();
    }
}
```



Handle Alerts

Alert Nedir?

Alert kullanıcıya bir tür bilgi vermek veya belirli bir işlemi gerçekleştirmeye izni için ekran bildirimini görüntüleyen küçük bir mesaj kutusudur. Uyarı amacıyla da kullanılabilir.

HTML Alerts

Bir alert çıktığında sağ click ile inspect yapabiliyorsak html alert'dir ve extra bir işleme gerek yoktur.

Js Alerts

Js alerts inspect yapılamaz, ekstra işleme ihtiyaç vardır.

Click for JS Alert

Click for JS Confirm

Click for JS Prompt

1.Simple Alert : Bu basit alert ekranda bazı bilgiler veya uyarılar görüntüler. Ok denilerek kapatılır

2.Confirmation Alert : Bu onay uyarısı bir tür işlem yapma izni ister. Alert onaylanıyorsa OK, onaylanmıyorsa Cancel butonuna basılır.

3.Prompt Alert : Bu Prompt Uyarısı kullanıcıdan bazı girdilerin girilmesini ister ve selenium webdriver metni sendkeys ("input....") kullanarak girebilir.



Handle Alert Methods

- **accept()** => Bir uyarıda(alert) OK'ı tıklamakla aynı.

```
driver.switchTo().alert().accept();
```

- **dismiss()** => Bir uyarıda(alert) Cancel'ı tıklamakla aynı.

```
driver.switchTo().alert().dismiss();
```

- **getText()** => Uyarıdaki(alert) mesajı almak için.

```
driver.switchTo().alert().getText();
```

- **sendKeys("Text")** => Uyarı(alert) text kutusuna text göndermek için

```
driver.switchTo().alert().sendKeys("Text");
```



Handle Alert Class Work

- Bir class olusturun: **Alerts**
- <https://testcenter.techproeducation.com/index.php?page=javascript-alerts> adresine gidin.
- Bir metod olusturun: **acceptAlert**
 - 1. butona tıklayın, uyarıdaki OK butonuna tıklayın ve result mesajının “You successfully clicked an alert” olduğunu test edin.
- Bir metod olusturun: **dismissAlert**
 - 2. butona tıklayın, uyarıdaki Cancel butonuna tıklayın ve result mesajının “You clicked: Cancel” olduğunu test edin.
- Bir metod olusturun: **sendKeysAlert**
 - 3. butona tıklayın, uyarıdaki metin kutusuna isminizi yazın, OK butonuna tıklayın ve result mesajında isminizin görüntüülendiğini doğrulayın.



Handle Iframe

```
Elements Console Sources Network Performance Memory Application Security Audits
<div id="mceu_17" class="mce-container mce-toolbar mce-toolbar-first mce-stack-layout-item">
  > <div id="mceu_19" class="mce-toolbar-grp mce-container mce-panel mce-stack-layout-item" hidefocus="1">
  > <div id="mceu_27" class="mce-edit-area mce-container mce-panel mce-stack-layout-item" hidefocus="1" title="Rich Text Editor - ALT-0 for help" style="width: 100%; height: 100px; display: block;"> == $0
    > #document
      <!doctype html>
      > <html>
        > <head>...</head>
```

Iframe nedir?

- IFrame, bir web sayfasına içine yerleştirilmiş başka bir web sayfasıdır
- IFrame genellikle bir Web sayfasına dokuman, video veya interaktif media gibi başka bir kaynaktan içerik eklemek için kullanılır.
- <iframe> tag'ı bir inline frame belirtir.

<https://html.com/tags/iframe/>

driver.switchTo().f~

```
frame(WebDriver)
frame(int ... WebDriver)
frame.WebElement WebDriver
parentFrame WebDriver
defaultContent WebDriver
```



Handle Iframe

- Bir sayfada iframe varsa, Selenium bir iframe içindeki elementleri doğrudan göremez
- `switchTo()` metod'u ile iframe'e geçmenin 3 yolu vardır;

1.) index ile :

```
driver.switchTo().frame(index of the iframe); //index 0'dan baslar
```

2.) id veya name value ile

```
driver.switchTo().frame("id of the iframe");
```

3) WebElement ile

```
driver.switchTo().frame(WebElement of the iframe);
```



Handle Iframe

Iframe'den cikmak icin 2 komut vardir

driver.switchTo().parentFrame(); 1 ust seviyedeki frame'e cikartir

driver.switchTo().defaultContent(); En ustteki frame'e cikmak icin kullanilir

Birden fazla iframe varsa gecislerde dikkatli olmak lazim. Gecisler

her zaman basit olamayabilir

<https://html.com/tags/iframe/>



Handle Iframe Class Work

- Bir class olusturun: IframeTest
- <https://the-internet.herokuapp.com/iframe> adresine gidin.
- Bir metod olusturun: iframeTest
 - “An IFrame containing....” textinin erisilebilir oldugunu test edin ve konsolda .
 - Text Box'a “Merhaba Dunya!” yazın.
 - TextBox'in altında bulunan “Elemental Selenium” linkini textinin gorunur oldugunu dogrulayin ve konsolda yazdirin.



User Story:

- Sınıf : IFrame
- <https://testcenter.techproeducation.com/index.php?page=iframe>
- Ana sayfadaki 'An iframe with a thin black border:' metninde 'black border' yazısının olduğunu test edelim
- Ayrıca 'Applications lists' yazısının sayfada olduğunu test edelim
- Son olarak footer daki 'Powered By' yazısını varlığını test edilip



Handle Iframe ODEV

- Bir class olusturun: IframeOdev

1) <https://the-internet.herokuapp.com/iframe> sitesine gidiniz

2) sayfadaki toplam iframe sayısını bulunuz.

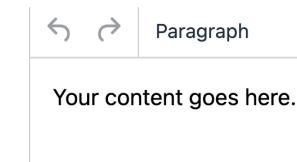
3) Sayfa basliginda ‘Editor’ yazını içerdigini test edelim.

4) Paragrafdaki yaziyi silelim

5) Sonrasında paragrafa “iframein icindeyim” yazisini yazdiralim

6) Alt kismdaki yazının ‘Elemental Selenium’ yazisini içerdigini test edelim

An iFrame containing the TinyMCE WYSIWYG Editor



Powered by Elemental Selenium



- How do you find the number of iframe on a page? Bir sayfadaki toplan iframe sayisini basil buluruz?
- ***Iframe ler iframe tagi kullanilarak oluşturulur. Bu durumda find elements by frame locatorini kullanarak tum iframeleri bulurum
- ***driver.findElements(By.tagName("iframe"));
- ***driver.findElements(By.xpath("//iframe"));

- Explain how you can switch back from a main page? Tekrar ana sayfaya nasil donersin?
- ***2 farkli yol vardır.
- ***defaultContent yada parentFrame metodlarını kullanarak gecis yapabiliriz.

- What is the difference between defaultContent and parentFrame? defaultContent ve parentFrame arasindaki farklar nelerdi?
*** Oncelikle her ikiside framein misina cikmamizi saglar. Ama defaultContent direk ana sayfaya atlatır, parentFrame ise bir üst seviyeye atlatır

- What might be the reason for your test case to fail? Bir test case in fail etme sebepleri neler olabilir?
 1. Yanlis locator. Cozum : Tekrar elementi locate etmek. Değişken element var ise ona gore dynamic bir xpath yazarız.
 2. Wait / Syncronization /Bekleme problemi: Cozum : implicit wait de yeterli sure oldugundan emin oluruz. O sureyi artıracaktır. 30 saniye verilebilir. Explicit wait kullanilabilir(sonra gore).
 3. Iframe. Cozum : switch to iframe
 4. Alert. Cozum : switch to alert
 5. New window(Yeni pencere) : switch to window

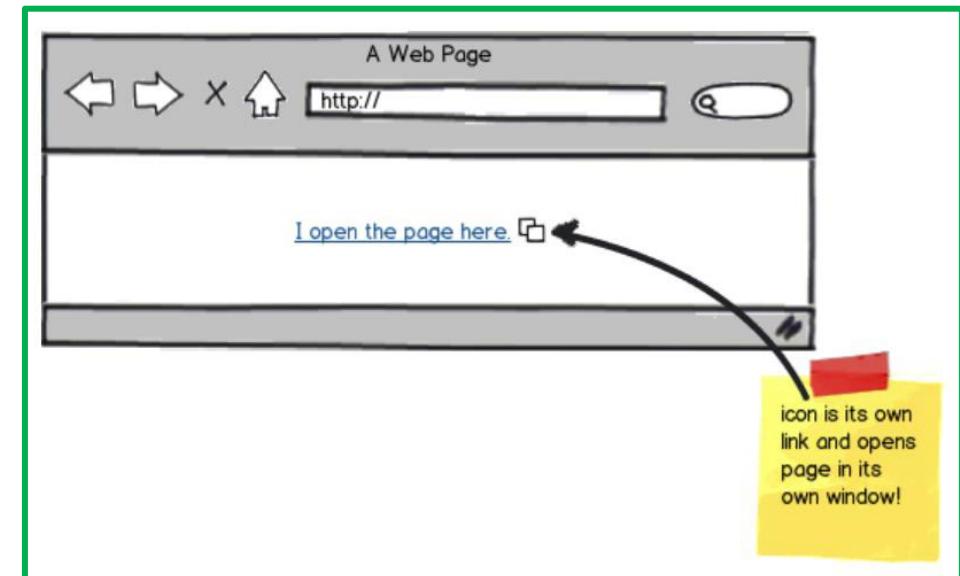


Windows



Handle Windows

- Bazen bir butona tıkladığımızda, başka bir sekmede(tab) yeni bir pencere açılır.
- Birden fazla pencereyle çalışırken driver'a pencereler arasında geçis yapmamız gereklidir.
- Pencereler arasında geçis yapmak için **window handle** değerini kullanırız.
- **window handle** : Selenium WebDriver'in, WebDriver objesi başlatıldığından her pencereye verdiği unique alfanumerik kimlik değeridir.





Handle Windows Class Work

- Tests package'ında yeni bir class olusturun: WindowHandle
- <https://the-internet.herokuapp.com/windows> adresine gidin.
- Sayfadaki textin “Opening a new window” olduğunu doğrulayın.
- Sayfa başlığının(title) “The Internet” olduğunu doğrulayın.
- Click Here butonuna basın.
- Acilan yeni pencerenin sayfa başlığının (title) “New Window” olduğunu doğrulayın.
- Sayfadaki textin “New Window” olduğunu doğrulayın.
- Bir önceki pencereye geri döndükten sonra sayfa başlığının “The Internet” olduğunu doğrulayın.



Handle Windows Method'ları

- 1.) İçinde olduğumuz sayfanın window handle değerini alma

```
driver.getWindowHandle();
```

- 2.) Pencereler arasında geçiş yapma(switch)

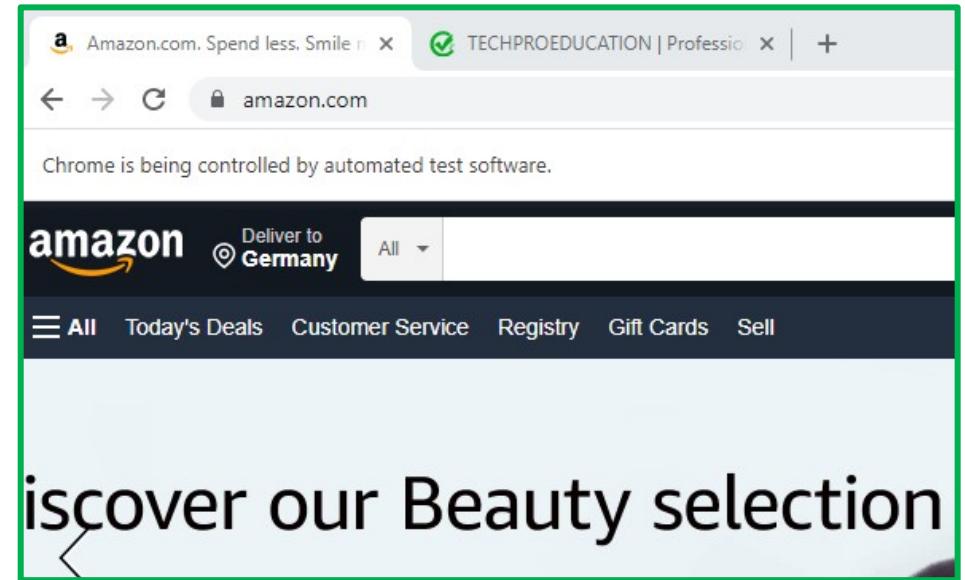
```
driver.switchTo().window(sayfa1HandleDegeri);
```

- 3.) Yeni tab oluşturup geçiş yapma(switch)

```
driver.switchTo().newWindow(WindowType.TAB);
```

- 4.) Yeni window oluşturup geçiş yapma(switch)

```
driver.switchTo().newWindow(WindowType.WINDOW);
```





Selenium 4 Windows

```
//      1. TechPro education sayfasinin titleinin "Techpro Education" metnini  
icerdigi test edin  
  
//      2. Yeni bir pencerede Amazon sayfasini acip titleinin "Amazon" metnini  
icerdigi test edin  
  
//      3. Yeni bir pencerede linkedin sayfasini acip titleinin "LinkedIn" metnini  
icerdigi test edin
```



JUnit

Handle Windows Homework

- <https://testcenter.techproeducation.com/index.php?page=multiple-windows>
- Title in ‘Windows’ oldugunu test edin
- Click here a tiklayin
- Sonra açılan sayfadaki title in ‘New Window’ oldugunu doğrulayın



Authentications-Güvenlik Doğrulamaları



Basic Authentication

the-internet.herokuapp.com/basic_auth

Authentication Nedir?

- Herhangi bir internet kullanıcısının, uygulamanın ya da programın, söz konusu sisteme dahil olup olamayacağını belirleyen doğrulama sistemini ifade eder.
- Uygulama ana sayfalarındaki kullanıcı adı ve password istemek de bir authentication'dır.
- Bu authentication'i yapabilmek için uygulamanın kullanıcılarına authentication'i nasıl yapacağına dair bilgilendirme yapmış olması gereklidir.
- Örneğin yandaki uygulama için authentication aşağıdaki gibi yapılabilir.

Oturum açın
<https://the-internet.herokuapp.com>

Kullanıcı adı:

Şifre:

Oturum açın **İptal**

`https://username:password@URL`



JUnit

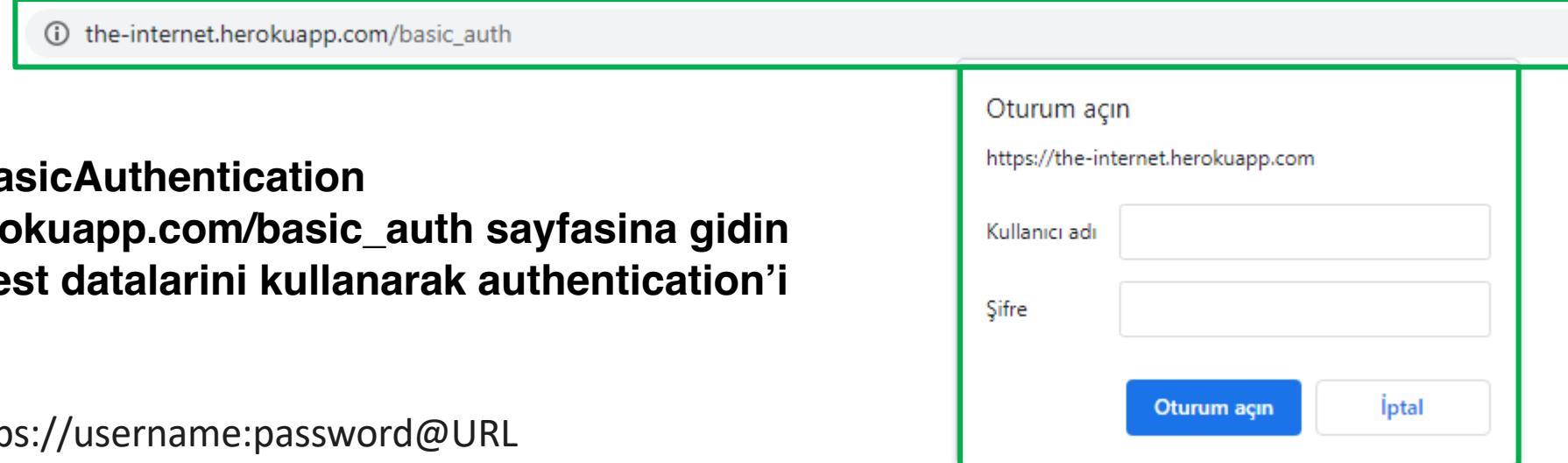
Basic Authentication Class Work

1. Bir class olusturun : **BasicAuthentication**
2. **https://the-internet.herokuapp.com/basic_auth** sayfasina gidin
3. asagidaki yontem ve test datalarini kullanarak authentication'i yapin

URL : https://username:password@URL
Username : admin
password : admin

4. Basarili sekilde sayfaya girin ve 'Congratulations!' yasizini doğrulayın

```
driver.get("https://admin:admin@the-internet.herokuapp.com/basic_auth");
```



The screenshot shows a browser window with a green border. The address bar contains the URL "the-internet.herokuapp.com/basic_auth". The main content area has a light gray background and displays the following text and form fields:

Oturum açın

https://the-internet.herokuapp.com

Kullanıcı adı:

Şifre:

Oturum açın **İptal**



Cookies- cerezler



Cookies- çerezler

Cookie nedir?

- Browserda gecici olusan bilgi dosyalarıdır
- Cookies bilgisayarımızda gecici olarak depolanır
- Cookies, kişisel bilgiler de dahil olmak üzere birçok bilgiyi içerebilir.
- Web siteleri, ancak sizin izin verdığınız bilgilere erişebilir.
- Bu web sayfaları, sizin vermediğiniz bilgilere erişemez ya da bilgisayarınızdaki diğer dosyaları görüntüleyemez.

Cookie leri automate edebiliriz:

➤ driver.manage(). method'u ile cookie'leri

- listeleyebilir
- isim ile çağırabilir
- Yeni cookie ekleyebilir
- Var olanları ismi silebilir
- Var olan tüm cookie'leri silebiliriz



driver.manage().cook

• addCookie(Cookie cookie)	void
• getCookies()	Set<Cookie>
• deleteAllCookies()	void
• deleteCookie(Cookie cookie)	void
• getCookieNamed(String name)	Cookie
• deleteCookieNamed(String name)	void

Ctrl+Down and Ctrl+Up will move caret down and up in the editor Next Tip



Cookies

- Class: Cookies
- Method: handleCookies
- Amazona git
- 1.toplam cookie sayisini bul
- 3. Bir Cookie yi smiyle bul
- 4. Yeni bir cookie ekle
- 5. Bir Cookie yi ismiyle sil
- 6. Tum cookie leri sil



ACTIONS



Actions Class

- Actions class I kullanarak **mouse and keyboard** işlemleri yapılır.
- 2 tip actions vardır:
 - **Mouse-based actions** : double-click, right click, click and hold, drag and drop, move over element etc, scroll.
 - **Keyboard-based actions**: Shift, Ctrl, and Alt keys.
- Actions has many useful mouse and keyboard functions





JUnit

Actions Class

- **1 . Adım:** Actions class'ta bir object oluşturulur.

Actions actions= new Actions(driver);

- **2 . Adım:** Üzerinde çalışmak istediğiniz WebElement öğesini bulunur.

WebElement element = driver.findElement(locator);

- **3 . Adım :** perform() kullanarak bu webelement üzerinde action gerçekleştirilir.
- Örneğin tıklamak için.

actions.click(element).perform();

NOT : 1. Action Class'ini her kullanmak istedigimizde yeniden obje olusturmamız gerekmekz. Aynı obje tekrar tekrar driver objesi gibi kullanılabilir
2. action objesi kullanılarak baslayan her komut, calismak icin **perform()** bekler.

```
driver.get("https://www.amazon.com");
Actions actions = new Actions(driver);

WebElement gununFirsatiElementi=driver.findElement(By.linkText("Today's Deals"));
actions.
    click(gununFirsatiElementi).
    perform();

Thread.sleep( millis: 3000);
WebElement amazonDevicesElement= driver.findElement(By.xpath("//span[text()='Amazon Devices'][2]"));

actions.
    sendKeys(Keys.PAGE_DOWN).
    perform();

Thread.sleep( millis: 3000);
actions.
    click(amazonDevicesElement).
    perform();
```



Mouse Base Actions Mouse Aksiyonları

- **doubleClick ()**: WebElement'e çift tıklama yapar
- **clickAndHold ()**: WebElement üzerinde click yapılı olarak bizden komut bekler.
- **dragAndDrop ()**: WebElement'i bir noktadan diğerine sürüklər ve bırakır
- **moveToElement ()**: Mouse'u istedigimiz WebElement'in üzerinde tutar
- **contextClick ()**: Mouse ile istedigimiz WebElement'e sağ tıklama yapar.



```
Actions actions = new Actions(driver);
actions.^
  • build() Action
  • sendKeys(WebElement target, CharSequence... keys) Actions
  • sendKeys(CharSequence... keys) Actions
  • perform() void
  • moveToElement(WebElement target) Actions
  • moveToElement(WebElement target, int ... offset) Actions
  • contextClick(WebElement target) Actions
  • contextClick() Actions
  • dragAndDropBy(WebElement source, int ... offset) Actions
  • dragAndDrop(WebElement source, WebElement target) Actions
  • clickAndHold(WebElement target) Actions
  • click() Actions
```

Press ⌘, to choose the selected (or first) suggestion and insert a dot afterwards. [Next Tip](#)



Mouse Base Actions Class Work

- Class: Actions1
- Me
- https://the-internet.herokuapp.com/context_menu sitesine gidin
- Kutuya sağ tıklayın
- Alert'te çıkan yazının “You selected a context menu” olduğunu test edin
- Tamam diyerek alert'i kapatın



Mouse Base Actions Class Work

- Class: Actions2
- Method : hoverOver() and test the following scenario:
 - Amazon a gidelim <https://www.amazon.com/>
 - Sag ust bolumde bulunan “Account & Lists” menüsüne git
 - “Account” secenegine tikka
 - Acilan sayfanin Title in “Your Account” icerdigini doğrula



Mouse Base Actions Class Work

- Class: ActionClass3
- Method : scrollUpDown()
- <https://techproeducation.com> a git
- Sayfanın altına doğru gidelim
- Sonra sayfanın üstüne doğru gidelim

- **PAGE_DOWN** =>scrolling down
- **PAGE_UP** =>scrolling up

- **ARROW_DOWN** =>scrolling down
- **ARROW_UP** =>scrolling up

- Arrow vs Page : Arrow moves less than Page



Mouse Base Actions Class Work

dragAndDrop(WebElement source, WebElement target) Actions

- Class: Action4
- Method: dragAndDropTest

- Given user is on <https://jqueryui.com/droppable/>
- And user moves the target element(Drag me to my target) in to destination(Drop here)



Keyboard Base Actions Klavye Aksiyonları

- Action Class'ından kullanacağımız method'lar ile klavyedeki tusları kontrol edebiliriz.
- Bunun için 3 method kullanıriz.
 - **sendKeys ()**: Öğeye bir dizi anahtar gönderir
 - **keyDown ()**: Klavyede tuşa basma işlemi gerçekleştirir
 - **keyUp ()**: Klavyede tuşu serbest bırakma işlemi gerçekleştirir





Mouse Base Actions Class Work

- Create a class: Action5
- Create a method keyboardActions
- Google a gidin <https://www.google.com>
- 'iPhone X fiyatları' yazdır (SHIFT TUSUNA BASILARAK)
- 'cok pahali!' Yazdır (SHIFT TUSUNA BASILMAYARAK)
- Ve ENTER tusuna bas

- We need to use Actions class to perform **mouse and keyboard** actions.
- There are two type actions:
 - **Mouse-based actions** : double-click, right click, click and hold, drag and drop, move over element etc, scroll.
 - **Keyboard-based actions**: Keyboard actions can function as **Shift**, **Ctrl**, and **Alt** keys.
- Actions has many useful mouse and keyboard functions



Mouse Base Actions Class Work

1- Actions nedir? Actions Class nedir ?

*Actions selenium dan gelen bir kütüphanedir. Mouse ve Keyboard islemlerini gerçekleştirmek için kullandığımız hazır bir selenium kutuphanesidir. Ornegin, sağ tıklamak(contextClick), çift tıklamak(doubleClick), elementin üzerine gitmek(moveToElement) gibi mouse islemlerini gerçekleştirebiliriz. Aynı zamanda, shift, control, delete, enter gibi keyboard tuslarınada actions yardımıyla basabiliriz. Bu tur islemlerde sendKeys metodu kullanılır.

2- Ne tur methodlar kullandın?

Actions metodları önemlidir. Ve tekrar tekrar kullanmam gerektiğinden, Actions metodlarını içeren Reusable metodları oluşturdum. Ve gerekiğinde bu reusable metodları kullanıyorum. Ornegin rightClick metodunu bir elemente raga tıklamak için test classımda vagıra bilirim.

3- Hangi metodları kullandın ne ne ise yarar?

Ornegin, sağ tıklamak(contextClick), çift tıklamak(doubleClick), elementin üzerine gitmek(moveToElement) gibi mouse islemlerini gerçekleştirebiliriz. Aynı zamanda, shift, control, delete, enter gibi keyboard tuslarınada actions yardımıyla basabiliriz. Bu tur islemlerde sendKeys metodu kullanılır.



Keyboard Base Actions Homework

Yeni Class olusturun ActionsClassHomeWork

- 1- "<http://webdriveruniversity.com/Actions>" sayfasina gidin
- 2- Hover over "Me First" kutusunun ustune gelin
3. Link 1" e tiklayin
4. Popup mesajini yazdirin
5. Popup'i tamam diyerek kapatin
6. "Click and hold" kutusuna basili tutun
- 7- "Click and hold" kutusunda cikan yaziyi yazdirin
- 8- "Double click me" butonunu cift tiklayin



- 1.Bir Class olusturalim KeyboardActions2
- 2.<https://html.com/tags/iframe/> sayfasina gidelim
- 3.video'yu gorecek kadar asagi inin
4. videoyu izlemek icin Play tusuna basin
5. videoyu calistirdiginizi test edin



Test01 :

- 1- amazon gidin
2. Arama kutusunun solundaki dropdown menunu handle edip listesini ekrana yazdırın
3. dropdown menude 40 eleman olduğunu doğrulayın

Test02

1. dropdown menuden elektronik bölümü seçin
2. arama kutusuna iphone yazıp aratin ve bulunan sonuç sayısını yazdırın
3. sonuc sayısı bildiren yazının iphone içerdigini test edin
4. ikinci ürüne relative locater kullanarak tıklayın
5. ürünün title'ni ve fiyatını variable'a assign edip ürünü sepete ekleyelim

Test03

1. yeni bir sekme açarak amazon anasayfaya gidin
2. dropdown'dan bebek bölümüne secin
3. bebek puset aratıp bulundan sonuç sayısını yazdırın
4. sonuç yazsının puset içerdigini test edin
- 5-üçüncü ürüne relative locater kullanarak tıklayın
- 6-title ve fiyat bilgilerini assign edelim ve ürünü sepete ekleyin

Test 4

- 1-sepetteki ürünlerle eklediğimiz ürünlerin aynı olduğunu isim ve fiyat olarak doğrulayın



FAKER CLASS



JAVA FAKER CLASS

- // fake data = mock data = gerçek olmayan test icin kullanan data
- // fake ssn, fake name, fake address
- // Fake data lar test caselerde data ihtiyacı oldugunda kullanilabilir
- // Java Faker kutuphanesi, fake data uretmek icin kullanilabilir

```
<dependency>
    <groupId>com.github.javafaker</groupId>
    <artifactId>javafaker</artifactId>
    <version>1.0.2</version>
</dependency>
```

```
//          1. Faker dependency pom a ekle
//          2. Faker objesi olustur
Faker faker = new Faker();
//          3. Fake data uret
String fName = faker.name().firstName();
```



Faker Kutuphanesi HOMEWORK

Faker class'i testlerimizi yaparken ihtiyav duyduğumuz isim, soyisim, adres vb bilgiler icin fake degerler uretmemize imkan tanir.

Faker degerler uretmek icin Faker class'indan bir obje uretir ve var olan method'lari kullaniriz.

1. "https://facebook.com" Adresine gidin
2. "create new account" butonuna basin
3. "firstName" giris kutusuna bir isim yazin
4. "surname" giris kutusuna bir soyisim yazin
5. "email" giris kutusuna bir email yazin
6. "email" onay kutusuna emaili tekrar yazin
7. Bir sifre girin
8. Tarih icin gun secin
9. Tarih icin ay secin
10. Tarih icin yil secin
11. Cinsiyeti secin
12. Isaretlediginiz cinsiyetin secili, diger cinsiyet kutusunun secili olmadigini test edin.
13. Sayfayı kapatın

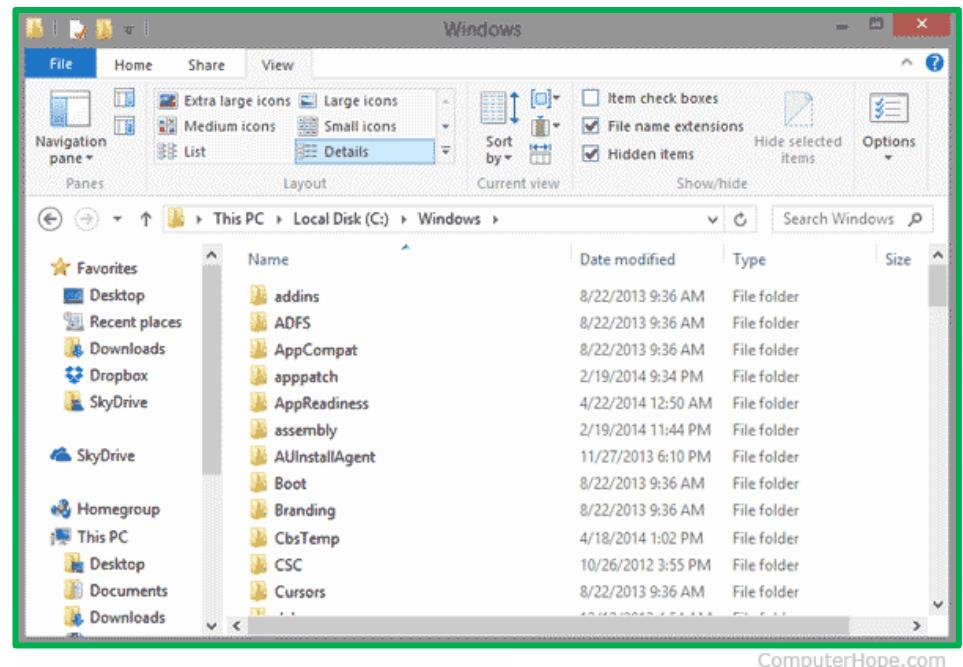


FILES-DOSYALAR



File Exist

- Selenium ile windows uygulamalarını test edemiyoruz. Ama test etmek için JAVA kullanabiliriz.
- Bilgisayarımızda bir dosya olup olmadığını(exist) kontrol etmek için Java'yı kullanabiliriz
 - **System.getProperty ("user.dir");** içinde bulunulan klasörün yolunu (Path) verir
 - **System.getProperty ("user.home");** bilgisayarımızda bulunan user klasörünü verir
 - **Files.exists (Paths.get (filePath));** Bilgisayarınızda dosyanın olup olmadığını kontrol eder
- İndirilen bir dosyanın indirme klasörümüzde olup olmadığını kontrol etmek için bu Java konseptini kullanabiliriz



ComputerHope.com



File Download/Exist Class Work

1.Class : FileExistTest

2.Method : isExist

3.Masa ustunden bir dosyanın varligini test et



File Download/Exist Class Work

1. Class : FileDownloadTest Method : downloadTestExist
2. <https://testcenter.techproeducation.com/index.php?page=file-download> adresine gidelim.
3. b10 all test cases dosyasını indirelim
4. Dosyanın başarıyla indirilip indirilmediğini test edelim



File Upload (Dosya Yukleme)

Dosya yükleme işlemini anlamak için önce manuel olarak test yapılmalı.

Daha sonra dosya, dosyanın bulunduğu yer (path) kullanılarak yüklenebilir.

<https://the-internet.herokuapp.com/upload>

File Uploader

Choose a file on your system and then click upload.

Dosya seçilmedi



File Upload Class Work

1. Tests packagenin altına bir class oluşturun :
C05_UploadFile
2. <https://the-internet.herokuapp.com/upload> adresine
gidelim
3. chooseFile butonuna basalim
4. Yuklemek istediginiz dosyayı secelim.
5. Upload butonuna basalim.
6. “File Uploaded!” textinin goruntulendigini test edelim.



Synchronization - Bekleme sureleri



JUnit

Synchronization - Selenium Waits

- Synchronization(Senkronizasyon), UI (kullanıcı arayüzü) otomasyon testi için çok çok önemlidir.
- Bir sayfanın uygulama sunucusu veya web sunucusu çok yavaşsa veya internet ağı çok yavaşsa, web sayfasındaki öğe (webelement) çok yavaş yüklenir.
- Bu durumda, komut dosyanız (test script) öğeyi bulmaya çalıştığında, öğeler yüklenmez.
- Bu yüzden test komut dosyası(test script) öğeyi bulamaz ve başarısız olur ve **NoSuchElementException** alırız.





Synchronization - Selenium Waits

- > Driver ile cihaz veya internet arasında yasanan senkronizasyon sorunlarını çözmek için sürücümüz(driver) belli şartlar ile bekletmemiz(wait) gereklidir.

1.) Java tabanlı wait

Thread.sleep : Javadan gelir ve kodları yazılan sure kadar bekletir. Sure dolduktan sonra alt satırdan işleme devam eder

2.) Selenium tabanlı wait'ler

Implicitly Wait: Sayfadaki tüm öğeler için global bir zaman aşımıdır(timeout).

Explicitly Wait: Çoğunlukla belirli öğeler için belirli bir koşul(expected condition) için kullanılır.





IMPLICITLY WAIT

Bir sayfanın yüklenmesi veya sayfadaki her bir öğenin locate edilebilmesi için driver'i bekletir.

Selenium tabanlı wait'lerde verilen sure max. bekleme süresidir, işlem daha önce biterse surenin bitmesi beklenmez, kod çalışmaya devam eder.

Genellikle otomasyon frameworklerinde olası senkronizasyon problemleri için default olarak implicitly wait ile kullanılır.

Implicitly wait'i TestBase class'ımızda kullanıyoruz.

```
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

Bu kod, driver'in sayfadaki herhangi bir weblement için maximum 10 saniye beklemesini istediğimiz anlamına gelir.



Webelement 10 saniyeden kısa sürede yüklenirse driver bulur ve devam eder. Örneğin, Webelement 3 saniye içinde yüklenirse, driver sadece 3 saniye bekleyecek ve bir sonraki satırı geçecektir.

Webelement 10 saniye içinde yüklenmezse, test case başarısız olur ve **NoSuchElementException**



JUnit

EXPLICIT WAIT

Beklenen bir durum(expected condition) olduğunda explicit wait kullanılır.

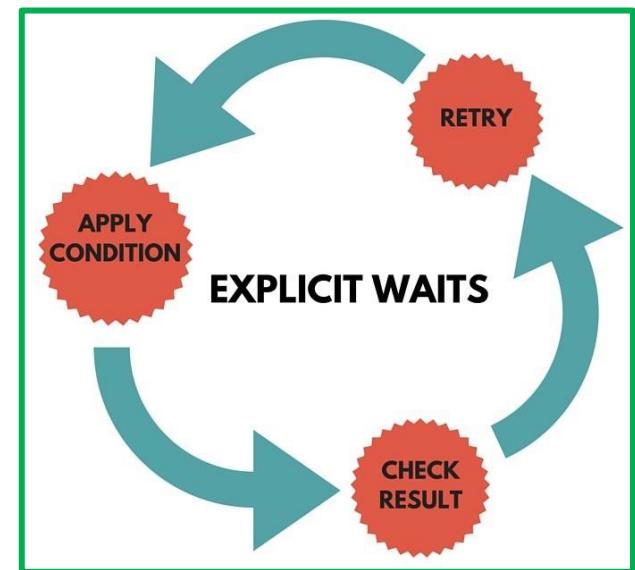
Implicitly wait ile cozulebilecek durumlar için explicitly wait kullanımına ihtiyac yoktur.

Nadiren karsımıza çıkan ve daha fazla bekleme süresi gerektiren belirli öğeler(web element) için explicitly wait kullanılır.

İlk olarak belirli miktarda bekleme süresi ile wait object create edilir.

```
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(20));
```

Oluşturduğumuz wait objesi sayesinde WebDriverWait class'ından until() method'u ile birlikte ExpectedConditions Class'ını kullanabiliriz. Örneğin WebElement'i locate etmek için wait object'i kullanma



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



EXPLICIT WAIT

Expected Conditions

- 1. **alertIsPresent()**
- 2. **elementSelectionStateToBe()**
- 3. **elementToBeClickable()**
- 4. **elementToBeSelected()**
- 5. **frameToBeAvailableAndSwitchToIt()**
- 6. **invisibilityOfTheElementLocated()**
- 7. **invisibilityOfElementWithText()**
- 8. **presenceOfAllElementsLocatedBy()**
- 9. **presenceOfElementLocated()**
- 10. **textToBePresentInElement()**
- 11. **textToBePresentInElementLocate**
d()
- 12. **textToBePresentInElementValue()**
- 13. **titleIs()**
- 14. **titleContains()**
- 15. **visibilityOf()**
- 16. **visibilityOfAllElement**
s()
- 17. **visibilityOfAllElementsLocatedBy()**



EXPLICIT WAIT

Expected Conditions

https://the-internet.herokuapp.com/dynamic_loading/1

Start buttonuna tıkla

Hello World! Yazının sitede olduğunu test et



Selenium Exceptions



Selenium Exceptions

➤ Interview Questions

- Tell me 5 exception that you get.-5 adet selenium da aldigin exceptions dan bahset.
- What type of exceptions do you get in selenium? - Seleniumda ki align exceptions tiplerinden bahset
- What is your solution?-Cozum ne?
- **What type of waits do you use?- Hangi resit wait kullanirsın?**
- **Which wait do you prefer?- Hangi wait tercih edersin?**
- **Why do you prefer that wait?- Neden onu tercih edersin?**
- **How to you resolve synchronization issue?-Senkronizasyon problemini nasil cozersin?**

- **NoSuchElementException**
- **TimeOutException**
- **NullPointerException**
- **InvalidSelectorException**
- **StaleElementReferenceException**
- **InvalidElementStateException**
- **WebDriverException**
- **ElementNotInteractableException**
- **ElementClickInterceptedException**
- **InvalidArgumentException**
- **JavascriptException**
- **RemoteDriverServerException**



NoSuchElementException

➤ NoSuchElementException:

- -Locator is wrong. Yanlis locator
- -Element is not on the page because of the iframe or pop up or new window. Popup, yeni pencere, iframe,...
- -Loading issue-syntonization issue. Implicit wait is not enough. Sayfa dolmada problemler ve yavaslamalar
- -Element is hidden- sakli olan elementler

➤ Solution:

- -Make sure locator is correct. Double check the syntax
- -Do manual testing to make sure there is no iframe, popups, or any other blocked on the page
- -If locator is correct and if we used correct switch(iframe, new window,...) then apply wait. Maybe element is hidden or loading slower.
- -Sometimes Implicit wait may not be enough to handle specific issues. For example, when we test reservation functionality, after clicking save button, we had to put explicit wait to verify the success message on the popup window.
- -Try using explicit wait. If that doesn't work, then try thread.sleep as a last resort

```
//WAY 1- Thread.sleep(1000) - not recommended - lazy
```

```
//WAY 2- Explicit Wait - hardworking
```

```
WebDriverWait wait = new WebDriverWait(Driver.getDriver(), timeOutInSeconds: 10);
WebElement popupMessageElement = wait.until(ExpectedConditions.visibilityOf(hotelRoomsPage.popupMessage));
//Verify the message: HotelRoom was inserted successfully
//Asserting message
Assert.assertEquals(popupMessageElement.getText(), expected: "HotelRoom was inserted successfully");
```

➤ NoSuchElementException:

- Yanlis locator
- Popup, yeni pencere, iframe,...
- Sayfa dolmada problemler ve yavaslamalar
- Sakli olan elementler

➤ Solution:

- Locatorin doğruluğunu tekrar kontrol et
- Manual test ile iframe, yeni pencere, alert,... gibi elementlerin varlığını kontrol et.
- Bekleme problemi olabilir. Implicit wait bekleme süresini cozmuyor. Bu durumda explicit wait yada fluent wait kullanırız



TimeOutException

➤ TimeOutException:

- If you use explicit wait and wait time is not enough/locator still incorrect then **TimeOutException**
- Simply put, wait time is over but the command is not completed

➤ Solution:

- -Increase the wait time
- -Try using different wait. For example try using javascript executor wait for the page to load. This is also a dynamic wait
- -I have different explicit waits in my Reusable util class. I have javascript executor wait as well.
ReusableMethods.waitForPageToLoad([1](#));
- -Case by case, I prefer different explicit waits to maximize the performance of the automation script

➤ TimeOutException:

- Explicit wait kullanildiginda, ve element bulunamadiginda alınır.
- Explicit wait & yanlis locator-> timeout
- Explicit wait & doğru locator & sure yeterli degil -> timeout
- Explicit wait & dogru locator & sure yeterli & iframe var -> time out

➤ Solution:

- -Sureyi arttirmak
- -Farkli explicit kullanmak: wait visibilityOfElementLocated YERINE presenceOfElementLocated. Yada javascript executor da ki waiter sayfa geçişlerini beklemek icin kullanabilirim
- -Locatori tekrar kontrol etmek
- -Frameworkunde hazır reusable method lar var Bu durumda timeoutexception aldigimda hızlıca o metodlar yardımıyla problemi cozebiliyoruz



NullPointerException

➤ NullPointerException

- When a variable is created, but not instantiated, when pointer does not show any value then we get NullPointerException.
- It is common to initialize the Driver, they we get the exception. Driver driver; but never use driver=new ChromeDriver()
- For example, in my page object model, I create page objects in test classes but when I forgot to use new keyword to instantiate, then I get the exception

➤ Solution:

- Instantiate the variable

➤ NullPointerException

- Degiskenleri oluşturup instantiate etmediğimizde alırız(new kelimesini kullanmadığımızda)
- Ornegin WebDriver driver; deyip = new ChromeDriver() demezisek null pointer exception alırız

➤ Solution:

- Degiskeni instantiate et

Declared but never instantiated

```
LoginPage loginPage;  
DefaultPage defaultPage;  
HotelRoomsPage hotelRoomsPage;
```

```
@BeforeMethod  
public void setUp(){  
    Driver.getDriver().get(ConfigReader.getProperty("url"));  
  
    LoginPage.username.sendKeys(ConfigReader.getProperty("username"));  
    LoginPage.password.sendKeys(ConfigReader.getProperty("password"));  
}
```

The screenshot shows a terminal window with a red arrow pointing to the error message. The message indicates 1 test failed and 1 was ignored, with a total execution time of 8 seconds and 664 milliseconds. The log output shows the detection of the W3C dialect and the final error stack trace:

```
Tests failed: 1, ignored: 1 of 2 tests – 8 s 664 ms  
Nov 07, 2021 9:38:48 AM org.openqa.selenium.remote.http.HttpCommandExecutor execute  
INFO: Detected dialect: W3C  
  
java.lang.NullPointerException Create br
```



StaleElementReferenceException

- Stale Element means the element is no longer fresh. It is old and no longer available
- When the element is no longer fresh on the page and you try to use it
- When the element is detached from the DOM
- When the element is removed after session started
- Solution:
- Relocate the same element again. You can use a new variable name as well.
- **Example:**

```
WebDriverManager.chromedriver().setup();
WebDriver driver = new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.carettahotel.com/Account/Logon");
WebElement userName=driver.findElement(By.id("UserName")); //When locator is not correct, get NoSuchElementException
WebElement passWord=driver.findElement(By.id("Password"));
WebElement loginButton = driver.findElement(By.id("btnSubmit"));

driver.navigate().refresh();

StaleElementReferenceException. When we refresh, the elements are no longer attached to dom
userName.sendKeys("manager");
passWord.sendKeys("Manager1!");
loginButton.click();
```

- Stale Element = eski, kullanılamaz durumda, kullanım artık elverişli değil
- -Sayfayı yenilediğinizde (refresh())
- -Sayfada ileri geri gittimizde
- COZUM:
- Elementi tekrar bulmak

```
try{
    WebElement userName2=driver.findElement(By.id("UserName")); //When locator is not correct, get NoSuchElementException
    WebElement passWord2=driver.findElement(By.id("Password"));
    WebElement loginButton2 = driver.findElement(By.id("btnSubmit"));
    userName2.sendKeys( ...keysToSend: "manager");
    passWord2.sendKeys( ...keysToSend: "Manager1!");
    loginButton2.click();
} catch(Exception e){
    System.out.println(e.getMessage());
}
```



WebDriverException

- WebDriverException
- When you close the browser, but still want to use that in correctly
- When the versions of the driver and browsers are not compatible

- Solution
- Your design should be good. We should use singleton Driver to prevent this exception
- Keep drivers up to date that is compatible with the browser version

- WebDriverException
- Driver versiyonları ve browser versiyonları aynı veya yakın olmadığında alınır
- Driver(Browser) kapattıktan sonra tekrar kullanmak istediğiniz durumlarda alınabilir.

- Solution
- Framework designinin güzel kurulmuş olması gereklidir, örneğin Driver sınıfında birden farza driver.quit() olmamalı



Interview Questions

➤ Tell me 5 exception that you get/What type of exceptions do you get in selenium?

- Null pointer, no such element, time out, stale element reference, web driver

➤ What is your solution?

- Read the error message and learn what is the error and where it is.
- In general, error happens because of the timing issue, wrong locator, or iframe/windows/alerts
- Each error has specific solutions but in general, I debug my code, and do some research then execute the code one more time after the fix.

➤ What type of waits do you use?

- I use implicit or explicit wait

➤ Which wait do you prefer?

- Explicit wait works better in some cases. When I use explicit wait, and still see timeout exception, then there can be other issues such as incorrect locators.

➤ Why do you prefer that wait?

- I already have implicit wait in my Driver class, but I prefer explicit wait when I need to use an expected condition such as element is not visible.

➤ How do you resolve synchronization issue?

- I use implicit or explicit wait. In my driver class, I already have implicit wait, that handles most of the wait issues, but when it is not enough than I use explicit wait.

➤ Seleniumda aldigin 5 exception

- Null pointer, no such element, time out, stale element reference, web driver

➤ Genel olarak exception aldiginda bozum yolun nedir?

- Hata mesajini oku
- Genel olarak zamanlama, yanlis locator, iframe hatalari olabiliyor
- Hata mesajina gore exception I cozdukden sonra tekrar kodumu calistirim

➤ Seleniumda kullandigin waitler nelerdir?

- Implicit yada explicit yada fluent

➤ Hangi waiti tercih edersin?

- Test Base (Driver sinifinda) implicit wait kullanıyorum. Cogu problemi bozuyor Cozemedigi durumlarda explicit wait kullanıyorum.

➤ Neden Explicit wait?

- Explicit wait element e ozel yazilabilir. Yerel cozum gerektiginde explicit wait I kullanmak zorundayiz.

➤ Senkronize(bekleme) problemlerini nasil cozersin

- Selenium waitlerini kullanarak. Implicit wait driver imda var, ama yetmediği durumlarda explicit wait kullanırı. Bu yüzden ReusableMethod sınıfında tekrar tekrar kullanabileceğimiz explicit wait methodları var. Kolaylıkla explicit wait ile bekleme problemlerini cozebiliyoruz.



Ekran Görüntüsü Alma



JUnit

Ecran Goruntusu Alma

➤ Tester olarak ekran goruntusu neden alınır?

- Documantasyon icin
 - Gecen test caselerin geçtiğini kayda geçirmek icin
 - Kalan test caselerin de kaldigini/ bug bulunduğu kayda geçirmek icin

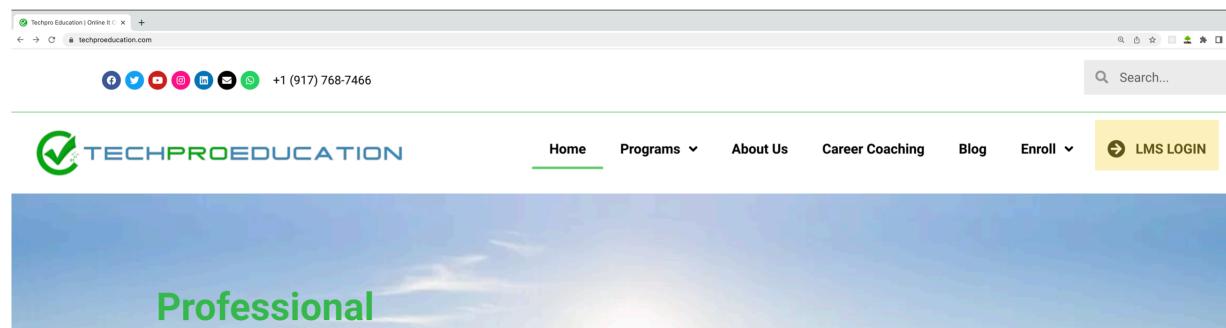


Ecran Goruntusu Manual Olarak Alma

- Manual test yaparken elle ekran goruntusualilip test execution dokumanina eklenir
- Word, ppt, yada excel tet raporlamada kullanabilir
- _____
- Techpro education sayfasina git
- _____
- LMS Login linkinin sayfada oldugunu test et
- _____
- MAC: Command +Shift + 4 -> Manual olarak ekran goruntusu al
- WINDOWS : Windows +Shift +S

➤ TEST EXECUTION DOCUMENTATION

1. Adim: ana sayfaya git ve LMS login elementini test et



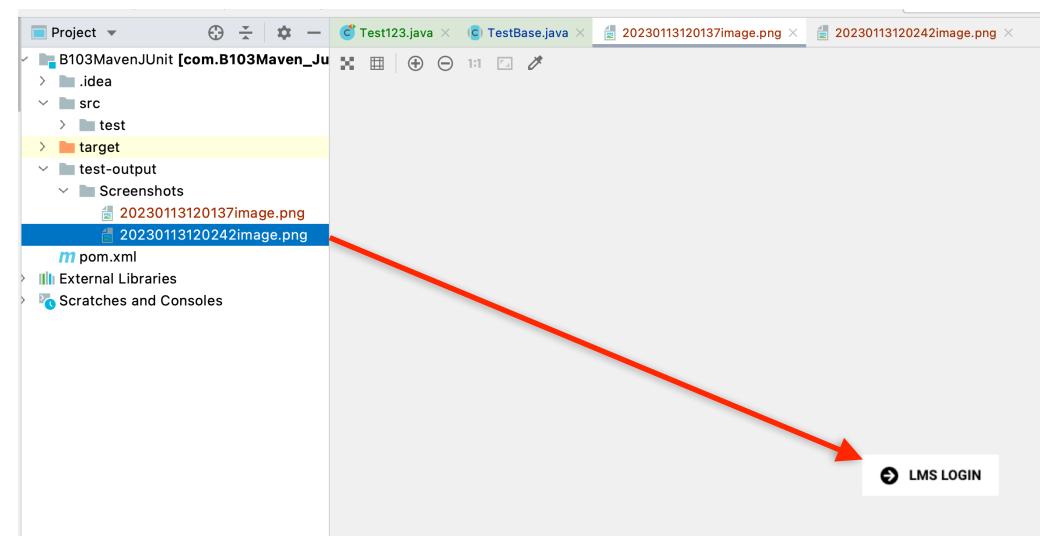
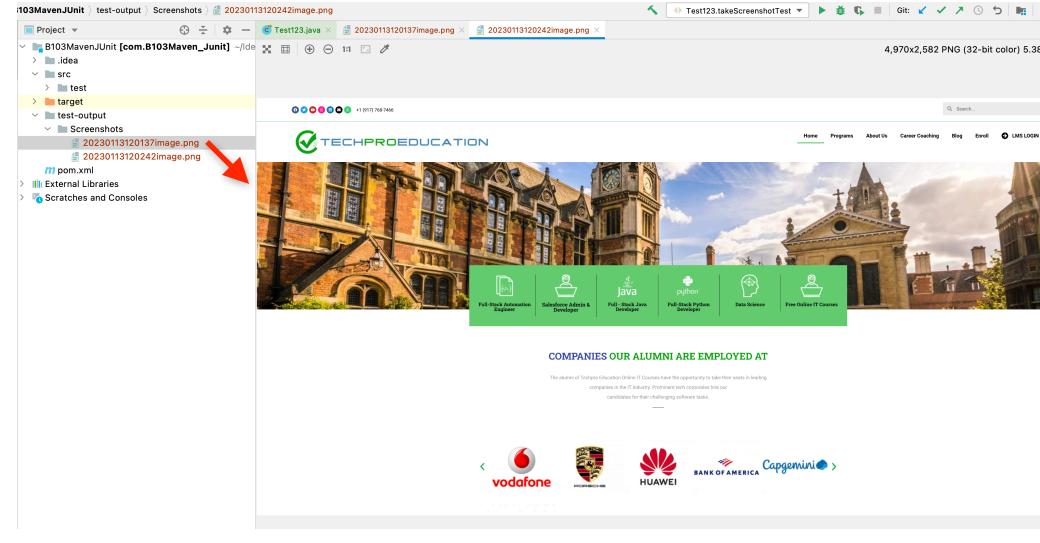
2. Documenta ettikten sonra JIRA YA bu dokuman alakali user story e yüklenir.



JUnit

Ecran Goruntusu Automasyon Ile Alma

- Atomosyon test yaparken selenium ile otomatik olarak ekran goruntusu alabiliriz
- Seleniumdan daki **TakeScreenshot** apiinden gelen **getScreenshotAs** metodu kullanılır
- Bu ekran goruntulerinin nereye kaydedileceğini, adının ne olacagini, ve hangi durumlarda(istedigimiz zaman, yada sadece fail etme durumunda,...) kaydedileceğini belirleyebiliriz
- Selenium 4 den itibaren tum ekran goruntusunu daya sadece belirli elementin ekran goruntusunu alabiliriz
- Reusable Method a ekleyeceğimiz metodlar yardımıyla bu iki islem kolaylıkla yapacağız





Get Screenshot / Tüm Sayfa

Selenium 3 ile tüm sayfanın fotoğrafını alabiliyorduk, Selenium 4 ile gelen güzel özelliklerden bir tanesi de
istedigimiz WebElement'in fotoğrafını almamıza imkan tanımı

1.Adım : Bir TakeScreenshot objesi olusturup driver'imizi TakeScreenshot'a cast yapalım

```
TakesScreenshot ts=(TakesScreenshot) driver;
```

**2.Adım : kaydettigimiz ekran görüntusunu projede istedigimiz yere
kaydedebilmek için path ile yeni bir File oluşturalım**

```
File tumResim= new File( pathname: "target/ekranResmi/tumSayfa");
```



**3.Adım : Takescreenshot objesini kullanarak getScreenshotAs() methodunu çalışıralım ve
gelen**

resmi geçici bir file'a assign edelim

```
File resimGecici=tss.getScreenshotAs(OutputType.FILE);
```

4.Adım : Kaydettigimiz görüntuyu, saklamak istedigimiz dosyaya kopyalayalım

```
FileUtils.copyFile(resimGecici,tumResim);
```



Get Screenshot / Spesific Webelement

1.Adım : Webelementi locate edelim

```
WebElement logoElementi=driver.findElement(By.id("nav-logo-sprites"));
```

2.Adım : kaydettigimiz ekran goruntusunu projede istedigimiz yere kaydedebilmek icin path ile yeni bir File olusturalim

```
File logoResmi=new File( pathname: "target/ekranResmi/element");
```



3.Adım : Webelement uzerinden getScreenshotAs() methodunu calistiralim ve gelen resmi olusturduğumuz gecici file'a assign edelim

```
File logoGecici=logoElementi.getScreenshotAs(OutputType.FILE);
```

4.Adım : Gecici resmi, saklamak istedigimiz dosyaya kopyalayalim

```
FileUtils.copyFile(logoGecici,logoResmi);
```



Get Screenshot Class Work

Class : screenShot1

- 1.Techpro education a git ve Sayfanin goruntusunu al**
- 2.-“QA” aramasi yap**
- 3.Acilen sayfanin metnini test et ve ekran goruntusu al: “Search Results for:
qa”**



Get Screenshot Class Work

Class : ScreenShot2

- 1.Techpro education a git**
- 2.Sosyal medya elemanlarını goruntusunu al**





Extent Reports



En sonki dersden neler hatırlıyoruz?

➤ Ekran goruntusu alma:

- getScreenShotAs metodu ile alınır. Bu metod Selenium dan gelir
- getScreenshotAs metodu TakesScreenshot api indan gelir
- Selenium 4 den itibaren 2 farklı sekilde ekran goruntusu alınabilir: Tum Sayfa, Özel element
- Oluşturmuş olduğumuz ReusableMetot yardımıza ister tum sayfanın, istersem belirli bir elementin ekran goruntusunu kolaylıkla alabilirim

➤ Extent Reports :

- Otomasyon raporları almak için kullanılır.
- Extent report ayrı bir API dir. Extent reports seleniumin disindada kullanilabilirler. Bu API dan gelen metodlar yardımıyla rapor sablonu oluşturabilir.
- 3 class kullandık
 - ExtentReports -> sablonu olusturu
 - ExtentHTMLReporter -> sablonu projeye ekler
 - ExtentTest -> raporlama islemiini yapar. Logları rapora yazdırır.



Extent Reports

➤ <http://www.extentreports.com/docs/versions/4/java/index.html>

➤ 3 onemli class

```
protected static ExtentReports extentReports;
protected static ExtentTest extentTest;
protected static ExtentHtmlReporter extentHtmlReporter;
```

```
ExtentReports extentReports = new ExtentReports(); //1. create object to set the location of the report
String filePath = System.getProperty("user.dir") + "/test-output/myprojectreport.html"; //create a custom report
ExtentHtmlReporter extentHtmlReporter = new ExtentHtmlReporter(filePath); //2. creating the report with the path
extentReports.attachReporter(extentHtmlReporter); //3. attaching the html report to our custom report
extentReports.setSystemInfo("Environment", "Environment Name");
extentReports.setSystemInfo("Browser", ConfigReader.getProperty("browser"));
extentReports.setSystemInfo("Automation Engineer", "ENGINEER INFORMATION");
extentHtmlReporter.config().setDocumentTitle("Crystalkey hoteReports");
extentHtmlReporter.config().setReportName("Crystalkey Automation Reports");
ExtentTest extentTest=extentReports.createTest("FCrystalkey Test", "Testing the Crystalkey functionality");
extentTest.info("Opening the URL");
Driver.getDriver().get(ConfigReader.getProperty("qa_environment"));
extentTest.pass("PASSED: SUCCESSFULLY COMPLETED THE REGISTRATION TEST");
```

The screenshot shows the Extent Reports dashboard with a single test run named 'Caretta Login Report'. The test run was executed on Nov 17, 2021, at 05:42:13 PM and passed. The details section lists four log entries:

Status	Timestamp	Details
✓	5:42:17 PM	Entering manager username
✓	5:42:17 PM	Entering password
✓	5:42:17 PM	Clicking on login button
✓	5:42:26 PM	Manager log in is successful



Extent Reports

- Extent reports CUSTOM HTML RAPORTLARI olusturmak icin kullanilir
- Bu raporlara özel proje veya
 - Sirket bilgisi
 - Takim bilgisi
 - QA bilgisi



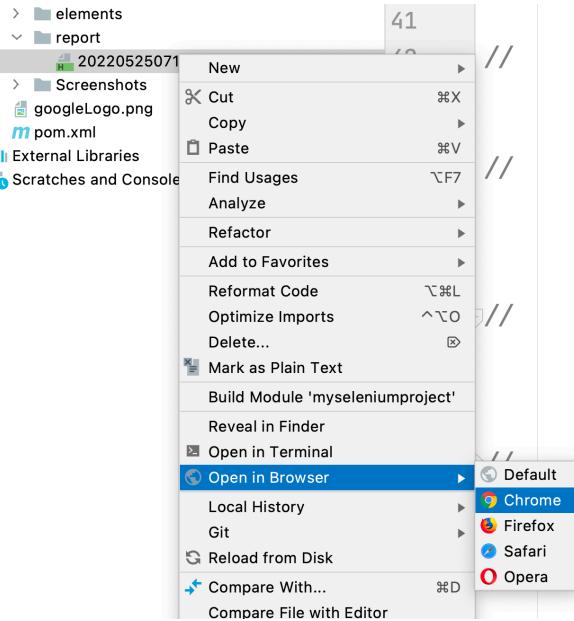
Extent Reports

- SET UP:
 - 1. Extent Reports dependency I maven projesine ekle
 - <https://mvnrepository.com/artifact/com.aventstack/extentreports>
 - 2. Create a class : Day_ExtentReports

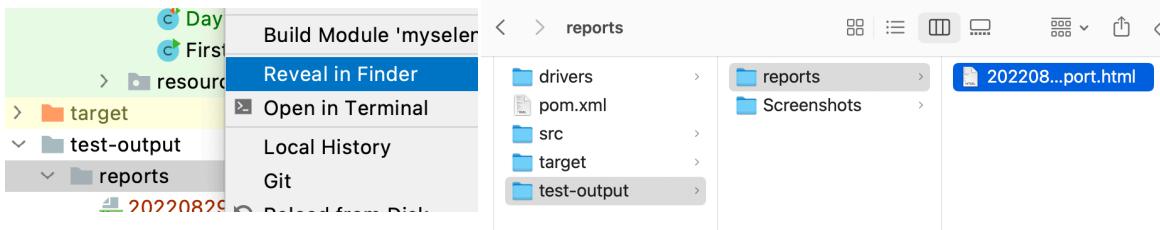


Extent Reports

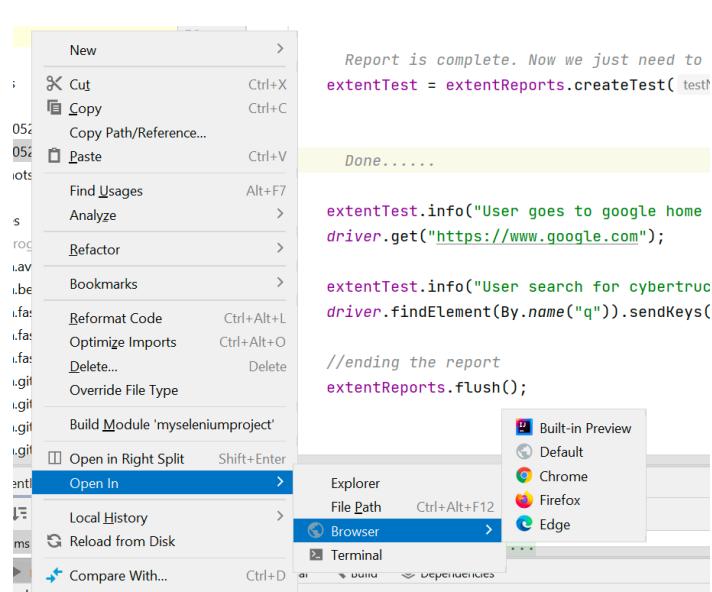
MAC raporu açma



MAC rapor dosyasına gitme

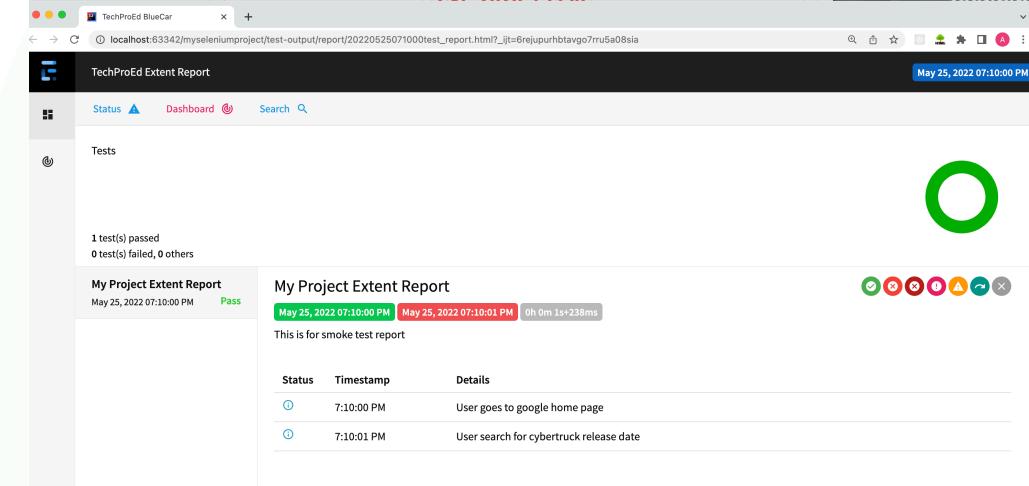


WINDOWS raporu açma

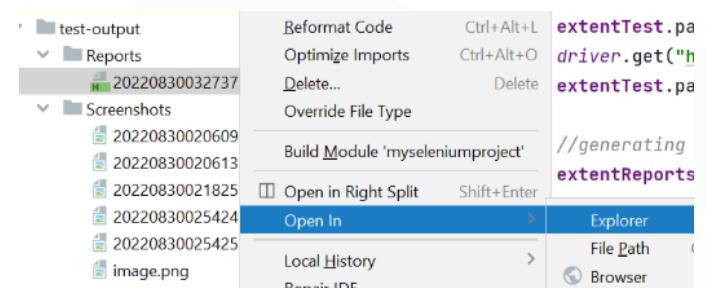


Report is complete. Now we just need to
extentTest = extentReports.createTest(test!)

```
extentTest.info("User goes to google home  
driver.get("https://www.google.com");  
  
extentTest.info("User search for cybertruck  
driver.findElement(By.name("q")).sendKeys(  
  
//ending the report  
extentReports.flush();
```



WINDOWS rapor dosyasına gitme





Extent Reports Interview Sorular

★ Raporlaramaları nasıl yaparsın?

- Manual testing yaparken elle ekran görüntüsü alıp work documanina eklerim. Sonra bu raporları JIRA ya user storye yüklerim
- Otomasyonda ise Extent Reports ile html raporlarını cikartip
- Extent Reports raporları değiştirilebilir, özel proje bilgiler eklenebilir raporlardır
- Projemde ki raporlama sistemi, test case fail ettiğinde otomatik olarak ekran görüntusunu alacak sekilde ayarladım

★ Extent Reports kullanmanın avantajları nelerdir? Neden Extent Reports i tercih ettiniz?

- Open source- Ücretsiz
- Farklı browserlar yada islerim sistemleriyle calisabilir
- Junit, testng, cucumber gibi farklı test otomasyon frameworkleriyle entegre olabilir
- Her bir test adimini log(rapora yazdırma) yapabiliyoruz
- Customizable-Editlenebilir

★ Extent reports ile raporlama işlemi nasıl gerçekleşir? Bilgi verirmisiniz?

- Oncelikle maven projeleri icin Extent reports dependency oluşturmamız gerekmektedir
- Reusable classimda extent reports ile alakalı reusable metodlarım var. Bu metodlar yardımıyla raporlarım oluşuyor.
- 3 onemli class vardır:
 - Extent reports
 - Extent Html Reporter. =====>>>> are user for generation the reports
 - Extent Tess
- Bu sınıflar yardımıyla, platform, browser, environment, tester, takim bilgileri gibi özel proje bilgileri ekleyebilirim. Ayrıca extent test objesi kullanarak pass, fail, skip gibi loglamalar yapabiliyorum.

★ Dependency lerini nerden aliyorsun?

- Şirkette internal bir repository miz var. Bu repodan ihtiyacım olacak dependencyi alıp kullanırııı.

★ Dependency şirket reposunda yoksa ne yapılır?

- İstedigim dependency icin istekde bulunuyorum. Managerlerin onayından sonra eklenen dependencyleri kullanıyorum.



WebTables



JUnit

HTML

Web Tables UI

ID	UserName	Email	Name Surname	BirthDate
2	manager	manager@st.com	manager	22.02.2020
3	customerservice	customerservice@st.com	customerservice	14.01.2020
154	123456	kristalhotel@gmail.com	cevdet oz	12.12.2020

The diagram illustrates the DOM structure of a table. The table is represented by a green box labeled "table". Inside the table, the `thead` section is shown with a red box, and the `tbody` section is shown with a green box. The `tbody` contains three rows, each represented by a blue box labeled "ROW". Each row contains several `td` elements, some of which are highlighted with colored boxes: a yellow box highlights the date "22.02.2020" in the first row; a blue box highlights the role "manager" in the second row; and a red box highlights the email "manager@st.com" in the second row. A red arrow points from the yellow box in the table to the corresponding `td>22.02.2020</td>` line in the DOM code. The entire table structure is enclosed in a green border.

BirthDate
22.02.2020
14.01.2020
12.12.2020

```
<table class="table table-striped table-bordered table-hover">
  <thead>
    <tr>...</tr>
  </thead>
  <tbody>
    <tr role="row" class="odd">
      <td>2</td>
      <td class="sorting_1">manager</td>
      <td>manager@st.com</td>
      <td>manager</td>
      <td>22.02.2020</td>
      <td>(545) 345-3453</td>
      <td>Manager</td>
      <td>...</td>
      <td>...</td>
    </tr>
    <tr role="row" class="even">
      <td>3</td>
      <td class="sorting_1">customerservice</td>
      <td>customerservice@st.com</td>
      <td>customerservice</td>
      <td>14.01.2020</td>
      <td>(543) 545-4354</td>
      <td>CustomerService</td>
      <td>...</td>
      <td>...</td>
    </tr>
    <tr role="row" class="odd">...</tr>
    <tr role="row" class="even">...</tr>
    <tr role="row" class="odd">...</tr>
  </tbody>
</table>
```



WEBTABLES

```
<table>           ======>>>table
                  <thead>           ======>>>header
                      <tr>           ======>>>table rows
                          <th>           ======>>>table header data
                          </th>
                      </tr>
                  </thead>
                  <tbody>           ======>>>table body
                      <tr>           ======>>>table rows
                          <td>           ======>>>table data-cell
                          </td>
                      </tr>
                  </tbody>
</table>
```

Amazon Music Stream millions of songs	Amazon Advertising Find, attract, and engage customers	Amazon Drive Cloud storage from Amazon	6pm Score deals on fashion brands
Sell on Amazon Start a Selling Account	Amazon Business Everything For Your Business	AmazonGlobal Ship Orders Internationally	Home Services Experienced Pros Happiness Guarantee
Audible Listen to Books & Original Audio Performances	Book Depository Books With Free Delivery Worldwide	Box Office Mojo Find Movie Box Office Data	ComiXology Thousands of Digital Comics
Goodreads Book reviews & recommendations	IMDb Movies, TV & Celebrities	IMDbPro Get Info Entertainment Professionals Need	Kindle Direct Publishing Indie Digital & Print Publishing Made Easy
Zappos Shoes & Clothing	Ring Smart Home Security Systems	eero WiFi Stream 4K Video in Every Room	Neighbors App Real-Time Crime & Safety Alerts



WEBTABLES

➤ <https://the-internet.herokuapp.com/tables>

- Create a class: WebTables
- Task 1 : Table1'i print edin
- Task 2 : 3. Row' datalarını print edin
- Task 3 : Son row daki dataları print edin
- Task 4 : 5. Column datalarini print edin
- Task 5 : İki parametreli bir Java metot oluşturalım: printData
 - Parameter 1 = row numarasi
 - Parameter 2 = column numarasi
 - printData(2,3); => 2nd satır, 3rd sutun daki datayı print etsin



Excel



Excel'in Yapısı

- Excel için daha önce inceledigimiz Web Table yapısına benzer bir yapı vardır.
- Excel'de bir hücredeki bilgiye ulaşmak için dosya/sayfa/satır/sutun sırasıyla ilerlemeliyiz
- Excel ile ilgili otomasyonda web table'da olduğu gibi sutun yapısı yoktur, ihtiyaç duyarsak satır ve hücre kombinasyonuya sutunlara gidebiliriz

Workbook excel dosyamız

Sheet Her açık excel sekmesi (Sheet1, Sheet2, etc)

Row(satır) Java, yalnızca içerisinde veri varsa satırları sayar.
Default olarak, Java perspektifinden satır sayısı 0'dır

Cell (hücre) Java her satıra bakar ve yalnızca hücrede veri varsa hücre sayısını sayar.

★ Workbook > Worksheet > Row > Cell

	A	B	C	D
1	Ülke (İngilizce)	Başkent (İngilizce)	Ülke (Türkçe)	Başkent (Türkçe)
2	Afghanistan	Kabul	Afganistan	Kabil
3	Albania	Tirana	Arnavutluk	Tiran
4	Algeria	Aljiers	Cezayir	Cezayir
5	Andorra	Andorra la Vella	Andorra	Andorra la Vella
6	Angola	Luanda	Angola	Luanda
7	Antigua & Barbuda	Saint John's	Antigua ve Barbuda	Saint John's
8	Argentina	Buenos Aires	Arjantin	Buenos Aires
9	Armenia	Yerevan	Ermenistan	Erivan
10	Australia	Canberra	Australya	Canberra
11	Austria	Vienna	Avusturya	Viyana
12	Azerbaijan	Baku	Azerbaycan	Bakü
13	The Bahamas	Nassau	Bahamalar	Nassau
14	Bahrain	Manama	Bahreyn	Manama
15	Bangladesh	Dhaka	Banglades	Dakka
16	Barbados	Bridgetown	Barbados	Bridgetown
17	Belarus	Minsk	Belarus	Minsk
18	Belgium	Brussels	Belçika	Brüksel
19	Belize	Belmopan	Belize	Belmopan
20	Benin	Porto Novo	Benin	Porto Novo
21	Bhutan	Thimphu	Butan	Thimphu
22	Bolivia	La Paz	Bolivya	La Paz
23	Bosnia & Herzegovina	Sarajevo	Bosna-Hersek	Saraybosna
24	Botswana	Gaborone	Botswana	Gaboron
25	Brazil	Brasilia	Brezilya	Brasilia
26	Brunei	Bandar Seri Begawan	Brunei	Bandar Seri Begawan
27	Bulgaria	Sofia	Bulgaristan	Sofya



Apache POI

- Apache POI, microsoft ofis dokumanlarına erişmek için kullanılan Java API'ıdır.
- Poi.apache.com sayfasından official dokumanlar incelenebilir.
- Excel kullanmak için poi dependencylerine ihtiyaç vardır;

```
<dependency>

    <groupId>org.apache.poi</groupId>

    <artifactId>poi-ooxml</artifactId>

    <version>4.1.2</version>

</dependency>

<dependency>

    <groupId>org.apache.poi</groupId>

    <artifactId>poi</artifactId>

    <version>4.1.2</version>

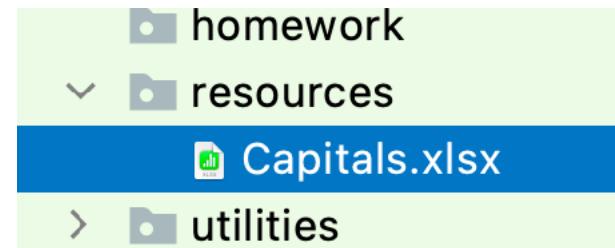
</dependency>
```



ADIMLAR

- Java dosyasına sağ tıkla ve resources adında bir klasör oluştur
- Excel dosyasını bu klasöre at

Resources dosyası, excel, word, text,.. gibi data dosyalarının olduğu klasörün adıdır





Read Excel Class Work

- 1.apache poi dependency'i pom file'a ekleyelim
2. Java klasoru altinda **resources** klasoru olusturalim
3. Excel dosyamizi resources klasorune ekleyelim
4. **excelAutomation** isminde bir package olusturalim
5. ReadExcel isminde bir class olusturalim
6. readExcel() method olusturalim
7. Dosya yolunu bir String degiskene atayalim
8. FileInputStream objesi olusturup,parametre olarak dosya yolunu girelim
9. Workbook objesi olusturalim,parameter olarak inputStream objesini girelim
- 10.**WorkbookFactory.create(inputStream)**
- 11.Worksheet objesi olusturun **workbook.getSheetAt(index)**
- 12.Row objesi olusturun **sheet.getRow(index)**
- 13.Cell objesi olusturun **row.getCell(index)**



Read Excel Class Work

Yeni bir test method olusturalim readExcel2()

- 1.satirdaki 2.hucreye gidelim ve yazdiralim
- 1.satirdaki 2.hucreyi bir string degiskene atayalim ve yazdiralim
- 2.satir 4.cell'in afganistan'in baskenti oldugunu test edelim
- Satir sayisini bulalim
- Fiziki olarak kullanilan satir sayisini bulun
- Ingilizce Ulke isimleri ve baskentleri bir map olarak kaydedelim



Write Excel Class Work

- 1)Yeni bir Class olusturalim WriteExcel**
- 2) Yeni bir test method olusturalim writeExcelTest()**
- 3) Adimlari takip ederek 1.satira kadar gidelim**
- 4) 5.hucreye yeni bir cell olusturalim**
- 5) Olusturdugumuz hucreye “Nufus” yazdiralim**
- 6) 2.satir nufus kolonuna 1500000 yazdiralim**
- 7) 10.satir nufus kolonuna 250000 yazdiralim**
- 8) 15.satir nufus kolonuna 54000 yazdiralim**
- 9)Dosyayı kaydedelim**
- 10) Dosyayı kapatalım**



JS Executor



Tekar

- Excel Okuma
 - Excelden veriyi okumak için hangi sırayla gitmem gereklidir
 - workbook > sheet > row > cell
 - Bir exceldeki top satır sayısı nasıl bulunur?
 - getLastRowNumber
 - Yada
 - getPhysicalNumberOfRows - toplam kullanılan satır sayısı
- Excel Yazdırma
- Microsoft dosyaları ile iletişime geçmek için apache poi dependency's kullanılır
- Su anki projende exceli nasıl kullandın?
 - Test caselerimde çeşitli datalar kullanıyorum. Örneğin kullanıcı adı, kullanıcı şifresi gibi dataları excelde tutuyoruz. Bu dataları test caselerde kullanmak için excelden çekiyorum ve test caselerimde kullanıyorum.
 - Dataları excelde tutmanın faydası, dataların derili toplu bir yerde olması, data güvenliği, data kayıplarına engel olmak.



Js Executors

★ What is JS Executor?

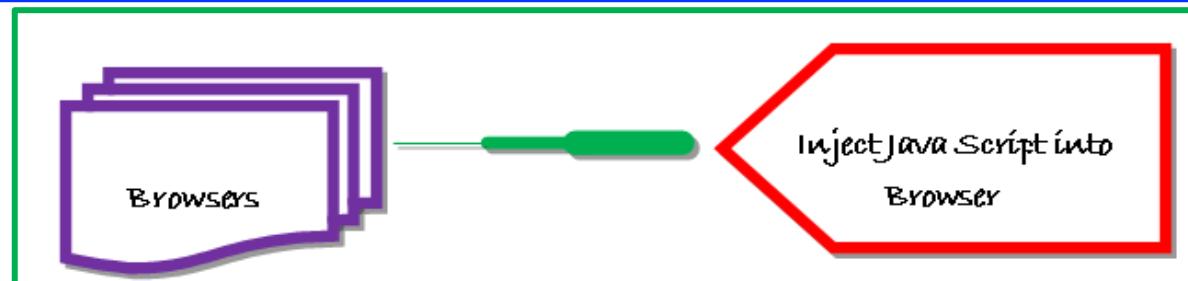
- Javascript Executor javascript kodlarını java frameworkinde kullanabilmek için oluşturulan bir API dir.
- Bu API seleniumda gelir.

★ Why you are learning javascript executor?

- Selenium ile web elementlerinin kontrollerini sağlarken selenium komutları yetersiz kalabilir veya sorunlarla karşılaşabiliriz. Bu sun da JS Executor kullanabiliriz. Click, belirli bir elemente scroll etme, type, input daki değerleri okuma... gibi çok farklı sekillerde işlem yapılabilir.
- JavaScript HTML kodlara direkt erişip yönetebilen bir script dili olduğundan bize çok fazla kolaylık sağlayabilir.

★ What we can do with JS executor?

- Click. Hidden elementlere tıklama
- Belirli elementlerin üzerine gitme, yada saga, sola, aşağı ya, yukarıya scroll etme
- Input lara yazı yazma
- Input lara değer atama
- Elementlerin renklerini değiştirme, arka rengini değiştirme, border(çerçeve) ekleme





JUnit

Js Executors

1.Adım : JavascriptExecutor kullanmak için ilk adım olarak driver’imizi JavascriptExecutor’a cast edelim

```
JavascriptExecutor jse=(JavascriptExecutor)driver;
```

2.Adım : Kullanacağımız WebElement’ı locate edelim

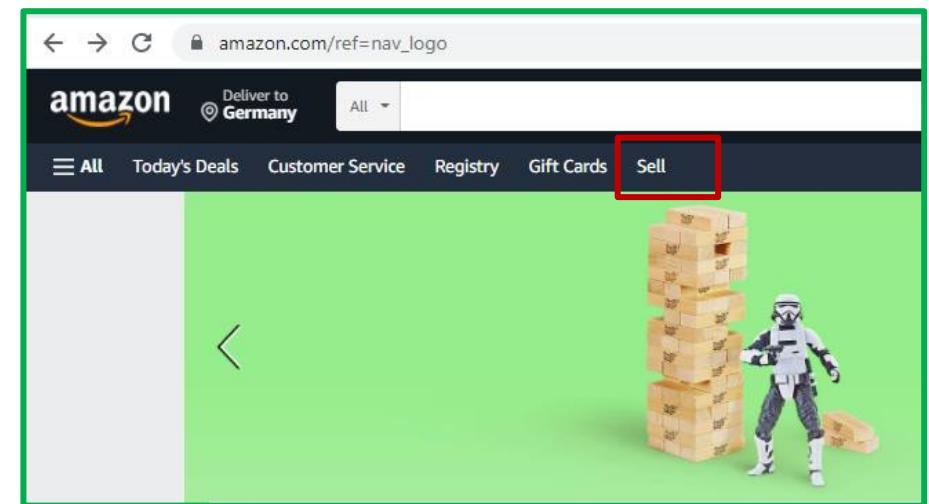
```
WebElement sellElementi= driver.findElement(By.xpath("//a[text()='Sell']"));
```

3.Adım : Js driver ile executeScript method’unu calistiralim, içine parameter olarak ilgili script ve webelement’ı yazalim

```
jse.executeScript( script: "ilgili script", ...args: "web element");
```

Ornegin Sell elementine click yapmak için

```
jse.executeScript( script: "arguments[0].click();",sellElementi);
```





JUnit

Js Executors ScroollIntoView

- 1)scrollIntoViewTest metodu olustur
- 2)Techpro education ana sayfasina git
- 3)We Offer elementi gorunur olacak sekilde uzerine scroll et ve ekran gorunturunu al
- 4)Ayni sayfada Enroll Free elementi gorunur olacak sekilde scroll et ve ekran goruntusunu al
- 5)Ayni sayfada WHY US? elementi gorunur olacak sekilde scroll et ve ekran goruntusunu al
- 6)Ayni sayfada tekrar Enroll Free elementi gorunur olacak sekilde scroll et ve ekran goruntusunu al

COURSES WE OFFER



Enroll Free Course

WHY US?



JUnit

Js Executors click

- 1)clickByJSTest
- 2)Techpro education ana sayfasina git
- 3)LMS LOGIN elementine tikla
- 4)Cikan sayfayinin URL inin “<https://lms.techproeducation.com/>” oldugunu test et



Js Executors SetAttribute

- 1) typeTest metodu olustur
- 2) Techpro education ana sayfasina git
- 3) Arama kutusuna QA yaz



Js Executors getAttribute

- 1)getValueTest metodu olustur
- 2)<https://www.priceline.com/> a git
- 3)Tarih kısmındaki Yazılı metinleri al ve yazdır



Js Executors getAttribute

- 1)getValueTest metodu olustur
- 2)<https://www.priceline.com/> a git
- 3)Submit butonunu rengini Kirmizi yaz



INTERVIEW SORULARI

1) Javascript Executor nedir?

1) Selenium'dan gelen ve javascript kodlarını calistirmaya yarayan bir interface dir. Normal selenium metotları yeterli olmadığı durumlarda kullanabilir.

2) Ne zaman ve nasıl kullandın?

1) Normal selenium metotları yeterli olmadığı durumlarda kullanabilir. Ornegin, bazı buttonlara tıklamada problem olabiliyor, yada belirli bir web elementin gorunur olacağı sekilde sayfanın kaydırma islemi, yada input kutularındaki değerleri almak için kullanabilir.

3) Sayfayı aşağı veya yukarı kaydırma islemi seleniumda nasıl yapılır?

1) Selenium'daki Actions yada javascript executor metotları ile kaydırma islemleri yapılabilir. Ornegin, actionslarda page_up, page_down, arrow_up, arrow_down metotları ile yada js executor'daki scrollintoview(true) ile yapılabilir. Ben oluşturduğumuz reusable metotlarını kullanarak bu islemlile kolaylıkla yapabiliyorum

4) Bir kutucukdaki(input) elemanın metni nasıl alınabilir?

1) Javascript executor ile alabiliriz. Cunki js executor ile attribute değerlerini alma imkanımız var. **getText() METORU BU DURUMLARDA CALISMAZ.**

5) Selenium da click yaparken problem yaşadığın oldumu?

1) Bazen yanlış locator, yada bekleme(wait) problemi gibi durumlarda problem yaşadım. Bazen gizli(hidden) elementlere tıklarkende problem yaşadım. Bu durumlarda javascript executor kullandım ve problemi cozdim.

6) Selenium atomayonunda ne tur problemler yaşadın?

1) TEKNİK PROBLEMLER : Yanlıs locator, bekleme(wait), alert, iframe, capche, pop-up şeklindeki reklam bildirimleri, multiple window, hidden elementlere tıklama, ...

2) TAKIMLA İLGİLİ PROBLEMLER: (Çok gündeme getirilmeme). Developer bazen bulduğum bunları kabul etmiyordu. Cozum: takımla beraber değerlendirdip fikir birliğine variyorduk.



Log4J



Log4j

➤ log4j nedir?

- Log4J bilgilerini **console** a yada bir **dosyaya loglamak(yazdırma)** icin kullanılan bir API dir
- Log4J **bilgi, hata,** yada **debug** mesajlarını yazdırma icin kullanılabilir
- **System.out.print** fonksiyonuna çok benzerdir, fakat kullanımı icin log4j dependency gerekmek dendir.

➤ Neden log4j kullanılır?

- Farkli log lama çeşitleri vardır
 - Fatal logs : `logger.fatal("This is fatal logger");`
 - Error logs : `logger.error("This is error logger");`
 - Informational logs : `logger.info("This is info logger");`
 - Debug logs : `logger.debug("This is debug logger");`
- Loglama anındaki zaman bilgisini alınabilir
- Dosyaya kalıcı olarak loglama işlemi yapılabilir



Log4j

The screenshot illustrates the configuration and execution of Log4j 2 in an IDE environment.

Project Structure: The project is named "myseleniumproject". The "resources" folder contains the configuration file "log4j2.xml".

log4j2.xml Content:

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="WARN">
    <Properties>
        <Property name="basePath">target/logs</Property>
    </Properties>
    <Appenders>
        <RollingFile name="file"
                     filePattern="${basePath}/TechProEd-%d{HH:mm:ss.SSS}.log">
            <PatternLayout>
                <pattern>%d %p %c{1.} [%t] %m%n</pattern>
            </PatternLayout>
            <SizeBasedTriggeringPolicy size="500" />
        </RollingFile>
        <Console name="Console" target="SYSTEM_OUT">
            <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>
        </Console>
    </Appenders>
    <Loggers>
        <Logger name="com.myfirstproject.utilities" level="trace" additivity="false">
            <AppenderRef ref="Console"/>
            <AppenderRef ref="file"/>
        </Logger>
        <Root level="fatal">
            <AppenderRef ref="Console"/>
        </Root>
    </Loggers>
</Configuration>
```

Execution Results: The "Demo1" tab shows the test results and logs.

Demo1 Tab:

- Tests passed: 1 of 1 test – 800 ms
- Output:

```
23:45:17.137 [main] DEBUG com.myfirstproject.utilities.Demo1 - This is debug logger
23:45:17.141 [main] INFO com.myfirstproject.utilities.Demo1 - This is info logger
23:45:17.141 [main] ERROR com.myfirstproject.utilities.Demo1 - This is error logger
23:45:17.141 [main] FATAL com.myfirstproject.utilities.Demo1 - This is fatal logger
```
- Process finished with exit code 0

Logs View:

- logs directory:
 - TechProEd-23:06:06.959.log
 - TechProEd-23:45:17.140.log
- test-classes directory
- com directory

Log Output:

```
2022-05-22 23:45:17,137 DEBUG c.m.u.Demo1 [main] This is debug logger
2022-05-22 23:45:17,141 INFO c.m.u.Demo1 [main] This is info logger
2022-05-22 23:45:17,141 ERROR c.m.u.Demo1 [main] This is error logger
2022-05-22 23:45:17,141 FATAL c.m.u.Demo1 [main] This is fatal logger
```



Log4j ayarları

- 1. Dependencileri ekle

```
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-api</artifactId>
    <version>2.17.2</version>
</dependency>
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>2.17.2</version>
</dependency>
```

- 2. Class olustur: Log4j

- 3. Logger object olustur:

```
private static Logger logger = LogManager.getLogger(Demo1.class.getName());
```

- 4. Logger fonksiyonuyla istenilen fonksiyonu kullan

```
logger.debug("Debug logger");
logger.info("Info logger");
logger.error("Error logger");
logger.fatal("Fatal logger");
```

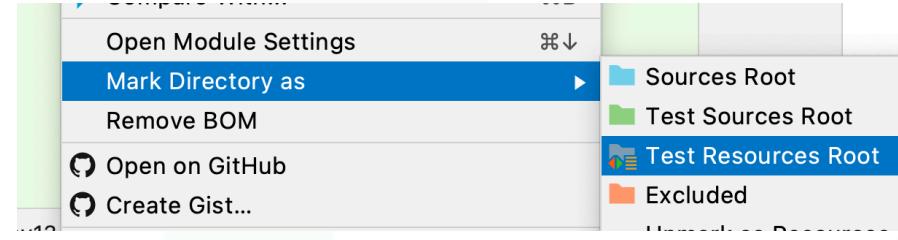
- 5. Consoldeki loglari kontrol edelim. Default olarak **error or fatal** gorunmus olacaktır



Log4j ayarları

- 1. Proje seviyesinde dosya olustur : resources
- 2. Resources in log4j2 xml dosyası olustur: log4j2.xml
- 3. Configuration dosyasına içeriği ekle. Diğer Slite: <https://logging.apache.org/log4j/2.x/manual/configuration.html>
- 4. Change the **level as trace** : <https://logging.apache.org/log4j/2.x/manual/customloglevels.html>
- 5. Run. Then you should see all logs

TO UPDATE THE FOLDER USE THIS:



Standard Log Level	intLevel
OFF	0
FATAL	100
ERROR	200
WARN	300
INFO	400
DEBUG	500
TRACE	600
ALL	<code>Integer.MAX_VALUE</code>



Log4j ayar dosyasi

We can do different configuration for different packages

Where the report file generates

where to show the logs such as console, file, ...
We should use this to log file on a file

formatting fileName pattern
for dynamic name generation

creates new file after the size exceeds.
Size can be changed

creates new file after the size exceeds.
Size can be changed

what format logs will be displayed

Logs the information.
Logger is used to override Root level configuration

When to log the logs: console, file,...

We use Logger for overriding root level configuration

All reports are coming from root level

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="WARN">
    <Properties>
        <Property name="basePath">target/logs</Property>
    </Properties>
    <Appenders>
        <RollingFile name="file"
            filePattern="${basePath}/TechProEd-%d{HH:mm:ss.SSS}.log">
            <PatternLayout>
                <pattern>%d %p %c{1.} [%t] %m%n</pattern>
            </PatternLayout>
            <SizeBasedTriggeringPolicy size="500" />
        </RollingFile>
        <Console name="Console" target="SYSTEM_OUT">
            <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>
        </Console>
    </Appenders>
    <Loggers>
        <Logger name="com.myfirstproject.utilities" level="trace" additivity="false">
            <AppenderRef ref="Console"/>
            <AppenderRef ref="file"/>
        </Logger>
        <Root level="error">
            <AppenderRef ref="Console"/>
        </Root>
    </Loggers>
</Configuration>
```

additivity="false" avoids duplicated messages
that can come from parent levels.
This should be used with Logger.

Levels that will be logged in the reports
trace=all levels



Log4j ayar dosyasi

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="WARN">
    <Properties>
        <Property name="basePath">target/logs</Property>
    </Properties>
    <Appenders>
        <RollingFile name="file"
            filePattern="${basePath}/TechProEd-%d{HH-mm-ss.SSS}.log">
            <PatternLayout>
                <pattern>%d %p %c{1.} [%t] %m%n</pattern>
            </PatternLayout>
            <SizeBasedTriggeringPolicy size="500" />
        </RollingFile>
        <Console name="Console" target="SYSTEM_OUT">
            <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>
        </Console>
    </Appenders>
    <Loggers>
        <Logger name="com.myfirstproject.tests" level="trace" additivity="false">
            <AppenderRef ref="Console"/>
            <AppenderRef ref="file"/>
        </Logger>
        <Root level="error">
            <AppenderRef ref="Console"/>
            <AppenderRef ref="file"/>
        </Root>
    </Loggers>
</Configuration>
```



Log4j2

➤ Log4j Nedir?

- Loglama yani yazdırma işlemi yapmak için kullanılan bir API dir
- Çeşitli seviyelerde loglama işlemi yapılabilir: uyarı, bilgi, debug, hata mesajları için kullanılabilir
- Javadaki `System.out.println` e benzer, ama `log4j` ile dosyaya detaylı şekilde yazdırma işlemi yapabilir.
- Şu anda projemde `log4j2` kullanmaktayım



Technical wellSkills - Yetenekler

➤ Diller(Languages):

- JAVA, Lamda, xml, SQL, HTML, CSS, ...

➤ Frameworks:

- JUnit(gorduk), TestNG(Basliyoruz), Cucumber(Ogrenecez)

➤ Design Pattern:

- Page Object Model(POM)

➤ Version Control System:

- Git, Bitbucket(ucretli, kara ile entegresi daha guzel)

➤ Front End Testing:

- Selenium WebDriver

➤ API Testing:

- Postman for manual testing, Restful for automation

➤ Backend/Database Testing:

- SQL, JDBC

➤ Continuous Integration(CI), Continuous Deployment(CD):

- Jenkins

➤ Remote Testing:

- Selenium Grid, Jenkins

➤ Mobile Testing:

- Appium

➤ Arac/Gerecler(Tools)

- Selenium WebDriver

- Maven

- JUnit, **TestNG**, Cucumber

- PostGreSQL

- JDBC

- IntelliJ, Eclipse, VS Code

- JIRA

- Jmeter

- Postman

- Restful API

- Jenkins

- Selenium Grid

- Git, Github, **Bitbucket**, AWC E2E

- Appium