



Selenium

Chapter 02



Neler Öğreneceğiz?



- Maven
- Create Maven Project
- JUnit
 - Annotations
 - Assertions
- Web Elements
- Inspect Page
- Web Elements Commands



Maven



- Maven, Java Tabanlı Projeler oluşturmak ve yönetmek için kullanılan bir araçtır.
- Temel olarak basit kelimelerle ifade etmek, Java Tabanlı Projeye bağımlılığı yönetmenin bir yoludur.
- Maven, büyük projeler üzerinde çalışırken POM (Project Object Model) ile proje oluştururken kullanılabilir.
- Birden çok IDE (IntelliJ, Eclipse, vs.) ile çalışır.
- Maven bir Java **derleme ve yönetim aracı**dır.
- Maven ile tüm otomasyon süreci sürecini yönetmek için **maven projesi** oluşturabilirsiniz.
- Maven, **pom.xml** dosyası ile dependency'leri yönetmemize yardımcı oluyor.
- Maven, **Java uygulamalarını derlememize, çalıştırmamıza ve dağıtmamıza** yardımcı oluyor.
- **Ant** ve **Gradle** gibi başka araçlar da var ancak **Maven en popüler** olanıdır.

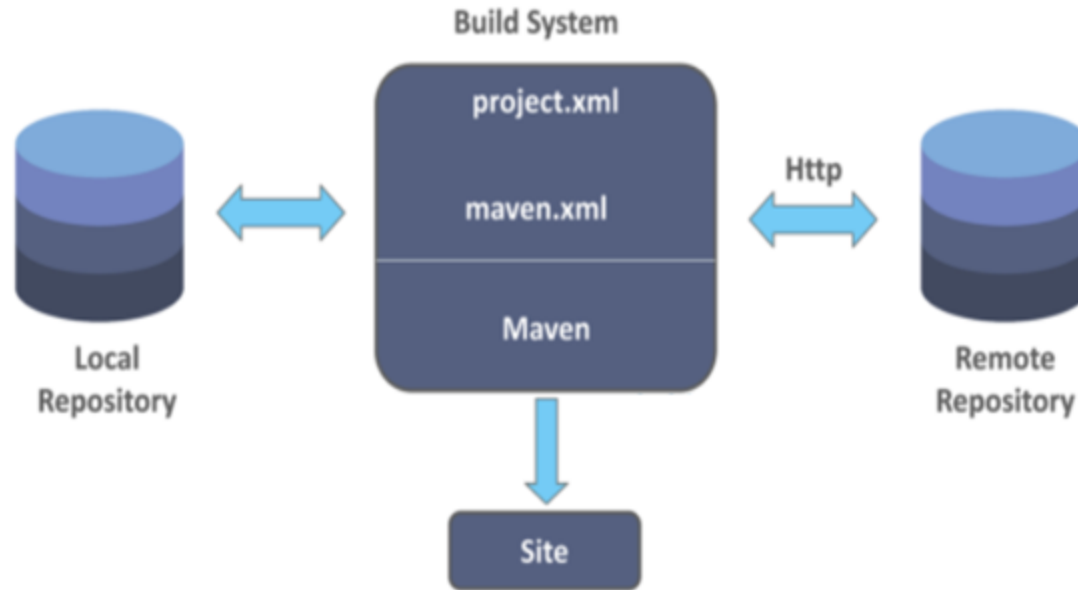


Maven is a Java build and management tool

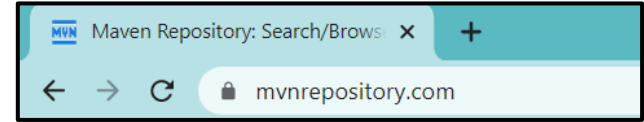
Maven Avantajları




Maven Mimarisi (Architecture)



- Aradığımız dependencies'lere kolayca erişmemizi sağlar.
- Bulduğumuz kütüphaneden en yeni ve en çok kullanılan sürümlerden en optimum olanı projemize ekleriz.



Home » [io.github.bonigarcia](#) » [webdrivermanager](#)

 **WebDriverManager**
Automated driver management and Docker builder for Selenium WebDriver

License: [Apache 2.0](#)

Tags: [io](#) [github](#)

Ranking: #1245 in MvnRepository (See Top Artifacts)

Used By: 328 artifacts

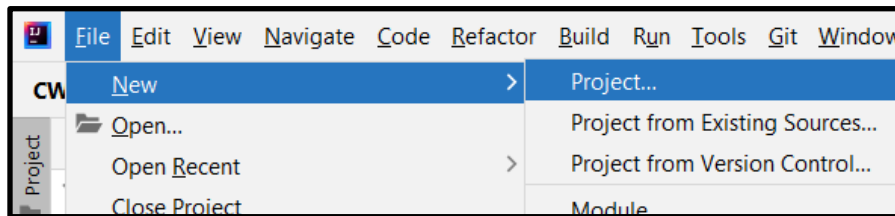
Central (73) [Liferay Public \(29\)](#)

	Version	Vulnerabilities	Repository	Usages	Date
5.2.x	5.2.1		Central	11	Jun, 2022
	5.2.0		Central	22	May, 2022
5.1.x	5.1.1		Central	21	Apr, 2022
	5.1.0		Central	41	Feb, 2022

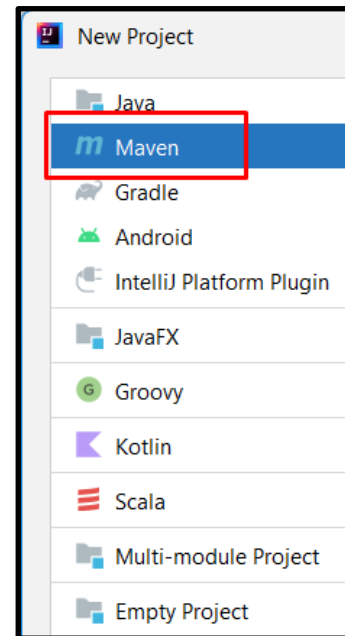
Create Maven Project



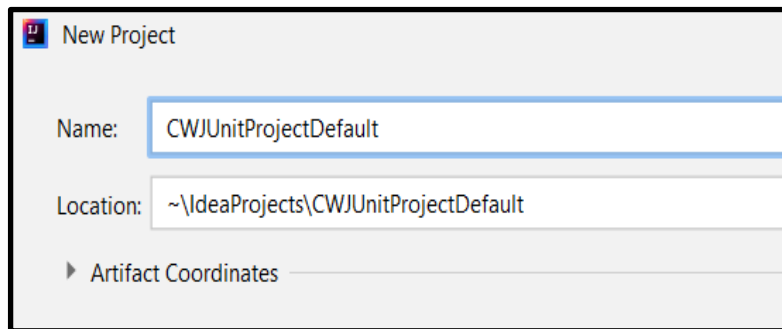
1



2



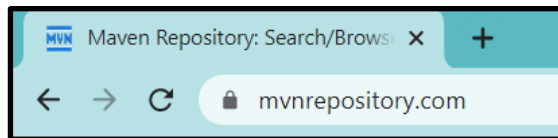
3



Create Maven Project



www.mvnrepository.com



4

5
pom XML dosyasına gidelim.

<properties> satırının altına **<dependencies>** **</dependencies>** tag'ı oluşturalım.

Bu aralığa kullanacağımız dependency'lerimizi ekleyeceğiz.

İlk projemizden **selenium- java** dependency'lerimizi alabiliriz.



Create Maven Project



6

The screenshot shows the Maven Central search results for the keyword "WebDriverManager". The search bar at the top contains "WebDriverManager" and a "Search" button. Below the search bar, it says "Found 4 results". There are two sorting options: "relevance" (selected) and "popular | newest". The first result is "1. WebDriverManager" by "io.github.bonigarcia » webdrivermanager". It has a GitHub icon and a description: "Automated driver management and Docker builder for Selenium WebDriver". The last release is "Jun 26, 2022". To the right of this result, it shows "328 usages" and an "Apache" license. The second result is "2. WebDriverManager" by "ru.sbtqa » webdrivermanager", with a gear icon and "3 usages" and an "Apache" license.

WebDriverManager aratalım (en güncel ve en çok kullanılan optimum versiyonu alalım)

Create Maven Project



```
Maven  Gradle  Gradle (Short)  Gradle (Kotlin)  SBT  Ivy  Grape  Leiningen  Buildr

<!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
<dependency>
  <groupId>io.github.bonigarcia</groupId>
  <artifactId>webdrivermanager</artifactId>
  <version>5.2.1</version>
</dependency>
```

7

Maven yazan bölümdeki kodları kopyalayalım.

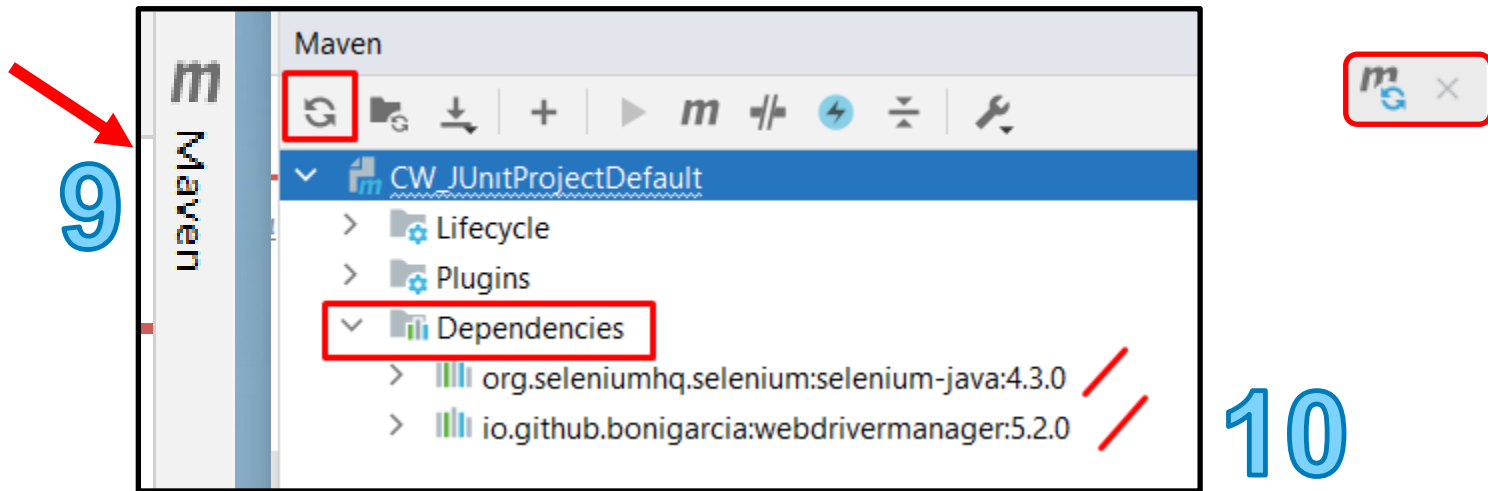
```
<dependencies>
  <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.3.0</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
  <dependency>
    <groupId>io.github.bonigarcia</groupId>
    <artifactId>webdrivermanager</artifactId>
    <version>5.2.0</version>
  </dependency>
</dependencies>
```

8

Kopyaladığımız kodları <dependencies> bölümüne yapıştıralım.

Create Maven Project



pom.xml dosyasının sağ tarafında **Maven** yazan bölümü tıklayalım.
Yenile butonuna tıklayıp aşağısındaki Dependencies bölümünün oluştuğunu ve altında eklediğimiz kütüphanelerin görüldüğünden emin olalım.

Create Maven Project



The screenshot shows the Maven Central search results for the query 'JUnit'. The search bar at the top contains 'JUnit' and a 'Search' button. Below the search bar, it states 'Found 2693 results'. The results are sorted by 'relevance', with options for 'popular' and 'newest'. The first result is 'JUnit Jupiter API' with 10,815 usages. The second result, which is highlighted with a red box, is 'JUnit' with 118,317 usages. This result includes the JUnit logo, the text 'junit » junit', a description 'JUnit is a unit testing framework for Java, created by Erich Gamma and Kent Beck.', and the release date 'Last Release on Feb 13, 2021'. The license 'EPL' is also indicated.

JUnit

Search

Found 2693 results

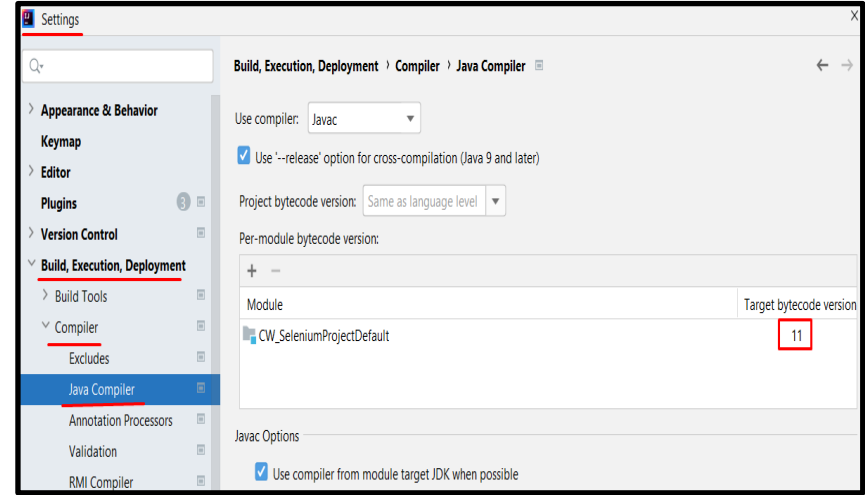
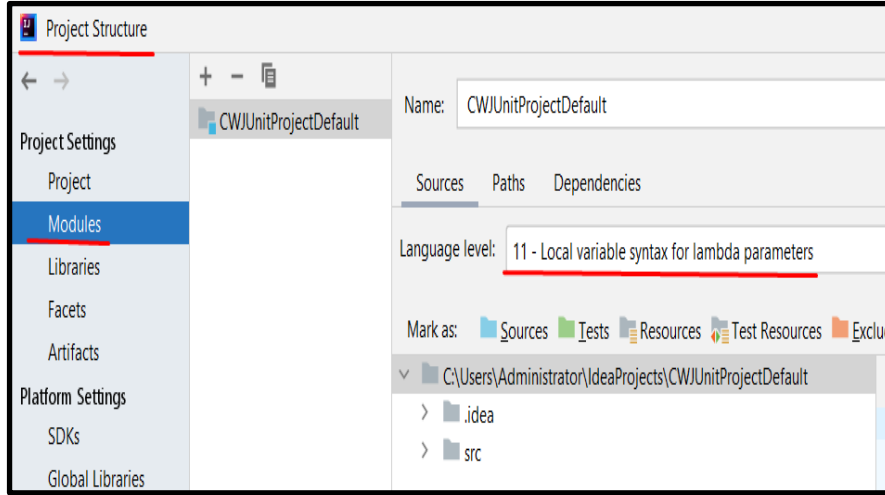
Sort: relevance | popular | newest

1. **JUnit Jupiter API** 10,815 usages
org.junit.jupiter » junit-jupiter-api
Module "junit-jupiter-api" of JUnit 5.
Last Release on Jul 4, 2022
EPL

2. **JUnit** 118,317 usages
junit » junit
JUnit is a unit testing framework for Java, created by Erich Gamma and Kent Beck.
Last Release on Feb 13, 2021
EPL

JUnit aratalım. (En güncel ve en çok kullanılan optimum versiyonu alalım)

Create Maven Project



Dikkat!

Aynı olmak zorundadır!

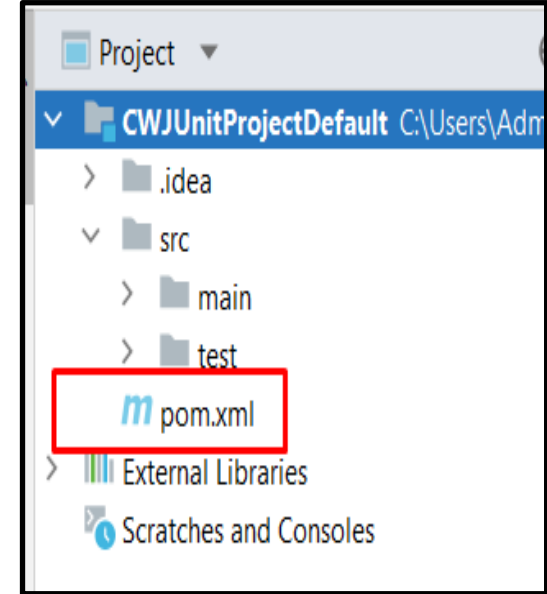
pom.xml



Maven bir araç (tool), pom bir xml dosyasıdır.
pom.xml Maven'in bir parçasıdır.

pom.xml'e yüklediğimiz dependencies ile projemizi rahatlıkla yönetebiliriz.

pom.xml: Maven'in kalbidir. Projemize eklemek istediğimiz kütüphaneleri POM xml'e dependency olarak ekleriz.



NOT:

POM, Project (Page) Object Model'in kısaltmasıdır.

pom.xml dosyası, bağımlılıklar, derleme dizini, kaynak dizini, test kaynak dizini, eklenti, hedefler gibi projeyi oluşturmak için maven için proje ve yapılandırma bilgilerini içerir.

Maven, pom.xml dosyasını okur ve ardından hedefi yürütür.

Maven 2'den önce proje.xml dosyası olarak adlandırılıyordu.

Ancak, Maven 2 (aynı zamanda Maven 3'te de) olduğundan, **pom.xml** olarak yeniden adlandırılır.

Dependencies



- pom dosyası, bir derleme aracı olan Maven'in ana eseridir.
- Maven'in birçok kullanımı vardır, ancak en yaygın olanı dependency yönetimidir. (yapı yönetimi de temel bir özelliktir)
- Çalışma şekli, projenizin ihtiyaç duyduğu dış kitaplıklara dependency'leri tanımlamanıza izin vermesidir.
- Maven'i kullandığınızda, bu kitaplıkları web' den (harici depolar) alır ve bunları yerleşik projenize ekler. Yani bu, dependency'lerin otomatik olarak işlenmesidir.
- pom file güncel tutmak için **mvnrepository.com** adresinden gerekli dosyaları bulur ve pom dosyamıza ekleriz.



Maven, project management tool'dur. Projelerimizi düzenli bir şekilde yönetmemize yardımcı olur.

Dependency: Kaynak kodlar olarak isimlendirebiliriz, istediğimiz kütüphaneleri bulunduğu GitHub adresinden otomatik olarak indirir.

mvnrepository.com: Aradığımız dependencies'e kolayca erişmemizi sağlar. Bulduğumuz kütüphaneden en yeni ve en çok kullanılan sürümlerden en optimum olanı projemize ekleriz.

Maven Project & WebDriverManager



```
System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");  
WebDriver driver = new ChromeDriver();
```



WebDriverManager

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





TIME FOR TASK

Use WebDriverManager

- Set Path.
- Create chrome driver.
- Maximize the window.
- Open google home page <https://www.google.com/>
- Verify that you are Google in the title.

So!
EASY

JUnit



- Açık kaynaklıdır.
- JUnit, en temel framework'tür.
- Software Tester'lar, otomasyon framework'ü oluşturmak için JUnit'i kullanabilir.
- Developerlar, unit framework oluşturmak için JUnit'i kullanabilir.
- Main ve class object oluşturmak zorunda değilsiniz.
- Test suite çalıştırabilir.



JUnit

Basically, JUnit is an open source unit testing tool and used to test small/large units of code. To run the JUnit test you don't have to create a class object or define the main method. **JUnit** provides assertion library which is used to evaluate the test result. Annotations of JUnit are used to run the test method. JUnit is also used to run the Automation suite having multiple test cases.

JUnit - Annotations



Annotation: Küçük bilgi içeren (Meta Data) yapılardır.

Notasyonlar genellikle Java'da konfigürasyon amacıyla kullanılır.

Kullanıldığı bölüme ekstra özellik katar. (Method Level - Class Level)

"@" işareti ile başlayan notasyonlardır.

1. **@Test** -> Marks a method as a TEST CASE.
2. **@Before** : Runs before EACH @Test annotation.
3. **@After** : Runs after EACH @Test annotation.
4. **@BeforeClass** : Runs before each class only once.
5. **@AfterClass** : Runs after each class only once.
6. **@Ignore** : Skipping a test case.

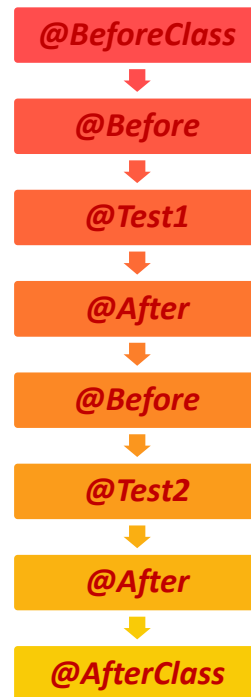
NOT:

Annotation'lar sayesinde main method olmadan testler execute edebilir.

Test method'u ve Test farklı yapılardır.

Test dediğimizde tek bir method veya içinde birçok class ve package barındıran bir yapı olabilir. Regression Test, Smoke Test vb.

Testmethod'u ise @Test notasyonu kullanılarak oluşturulan ve bağımsız olarak çalıştırabileceğimiz en küçük test yapısıdır.



JUnit - Assertions



JUnit Assertions kullanımı yazmış olduğumuz test senaryosunun sonunda elde edilen sonuç (**actual result**) ile beklenen sonuçların (**expected result**) karşılaştırılması yapılması için kullanılır. Bir test senaryosundan beklenen sonuç ile elde edilmiş olan **sonuç aynı** ise **test başarılı** olarak nitelendirilir.

assertEquals
assertTrue
assertFalse
assertNull
assertNotNull
assertSame
assertNotSame
assertArrayEquals

Example:

Assert.assertEquals("ABC", "ABC"); //Both the strings are equal and assertion will pass.

Assert.assertEquals("ABC", "DEF"); //Assertion will fail as both the strings are not equal.

Assert.assertEquals("Strings are not equal", "ABC", "DEF"); //message will be thrown if the equal condition is not satisfied.

If - Else Statement



Syntax:

Assert.assert....("excepted value", "actual value");



JUnit - Assertions



- Örneğin yazılan testler koşturma sırasında 3 test senaryosu geçti ve 2 test senaryosu doğru şekilde testten geçemediyse başarılı bir sonuç elde etmiş olmuyoruz.
- Assertions kullanımında birebir aynı sonuçların gelmesini beklememiz gerekmektedir.
- Bunlardan biri **Hard Assertions** diğeri ise **Soft Assertions**.
- Adından da anlaşılacağı gibi Hard Assertions; test senaryosu içerisinde geçen onaylama koşulu gerçekleştirilmeden bir sonraki test adımına geçilmez. Bunun için bir otomasyonda bir exception atılır. Exception atıldığında test durumu başarısız olarak sonuçlandırılır.
- Soft Assertions ise test senaryosu içerisinde bir onaylama koşulu gerçekleştirilme bile test adımına devam eder ve herhangi bir exception atmaz ve bir sonraki test senaryosu adımı ile devam eder.



assertEquals() – assertTrue()



assertEquals, beklenen sonucu gerçek sonuç ile karşılaştırmak için kullanılır. Beklenen sonuç ile gerçek sonuç eşit değil ise gerçekleştirilen test senaryosu sonucunda **assertionError** hatası fırlatır.

assertTrue, beklenen bir sonucun **true** olduğunun kabul edilmesi gerektiği zaman kullanılır. Parametre olarak iki değer alır. İlk parametre de bir mesaj gönderilir ikinci parametrede ise gönderilen mesajın doğruluğu için koşul belirlenir.

assertFalse() – assertNull() – assertNotNull()

assertFalse, beklenen bir sonucun false olması durumunda kullanılır. İki parametre alır. Parametrelerden biri mesajdır diğeri ise koşuldur. assertFalse ile koşul yerine getirilmez ise **assertionError** hatası fırlatır.

assertNull, beklenen bir sonucun null olup olmadığı kontrol edilmesi için kullanılır. Bir nesneyi parametre olarak alır ve nesne null değil ise **assertionError** hatası fırlatır.

assertNotNull, beklenen bir sonucun null olmadığını doğrulamak için kullanılır. Bir nesneyi parametre olarak alır ve nesne null ise **assertionError** hatası fırlatır.

assertSame() – assertNotSame() – assertEquals()



assertSame, parametre olarak verilen iki nesnenin aynı nesneye karşılık gelip gelmediğini kontrol eder. Eğer nesneler aynı nesneyi karşılamıyor ise `assertionError` hatası fırlatır.

assertNotSame, parametre olarak verilen iki nesnenin birbirine eşit olmadığını kontrolünü eder. Eğer aynı nesneye karşılık geliyor ise `assertionError` hatası fırlatır.

assertEquals, parametre olarak verilen iki dizinin eşit olup olmadığını kontrol eder. Her iki dizi içinde null değeri var ise bunlar eşit olarak kabul edilir. Eğer eşit değil ise `assertionError` hatası fırlatır.





Assertions

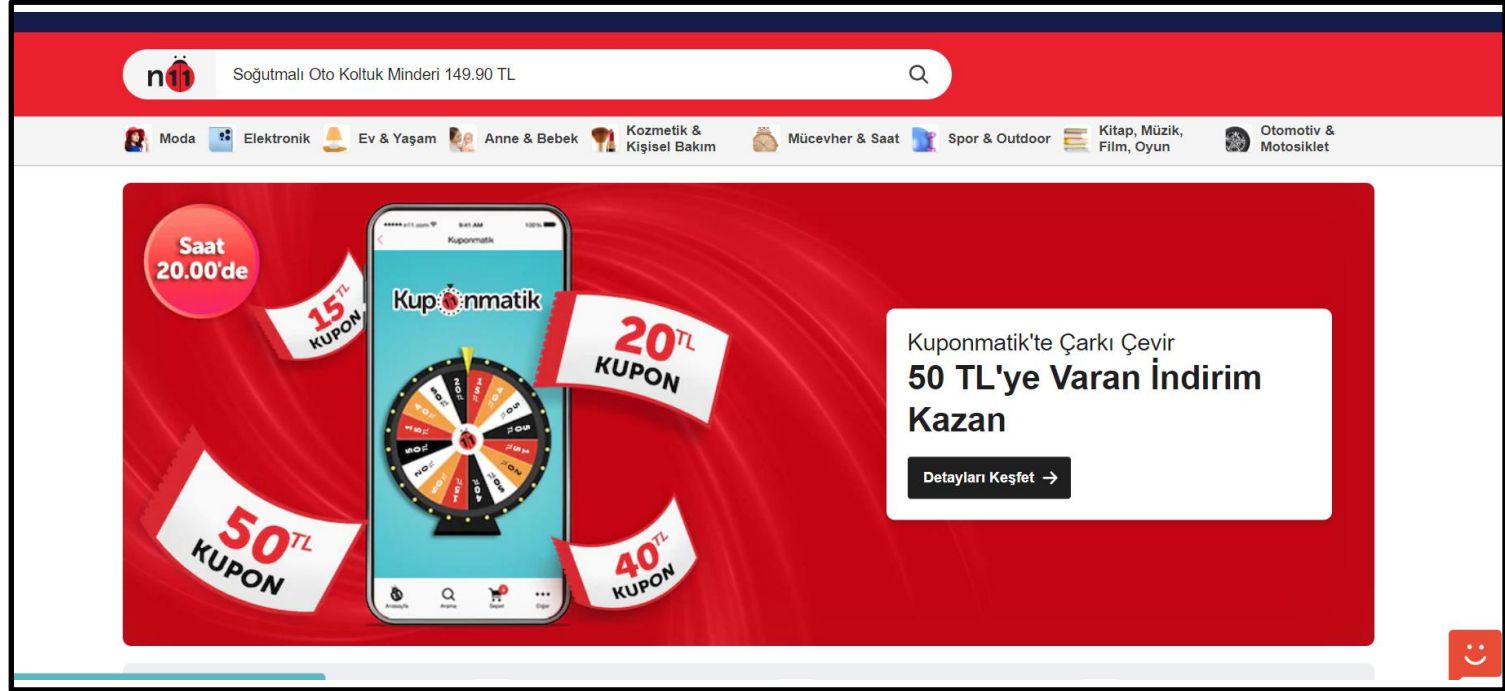
Go to Amazon homepage.

Do the following tasks by creating 3 different test methods.

- 1- Test if the URL contains Amazon.
- 2- Test if the title does not contain Facebook.
- 3- Test that the Amazon logo appears in the upper left corner.



WebElements



Web sayfasındaki elementleri bulma ve işlem yapmaya yarar.

WebElements



Bir Selenium WebElement, aslında bir web sitesindeki bir HTML ögesidir.

HTML belgeleri HTML öğelerinden oluşur.

Her HTML ögesi, bir başlangıç etiketi ve bir bitiş etiketinden oluşur. İçerik, etiketler arasında yer alır.

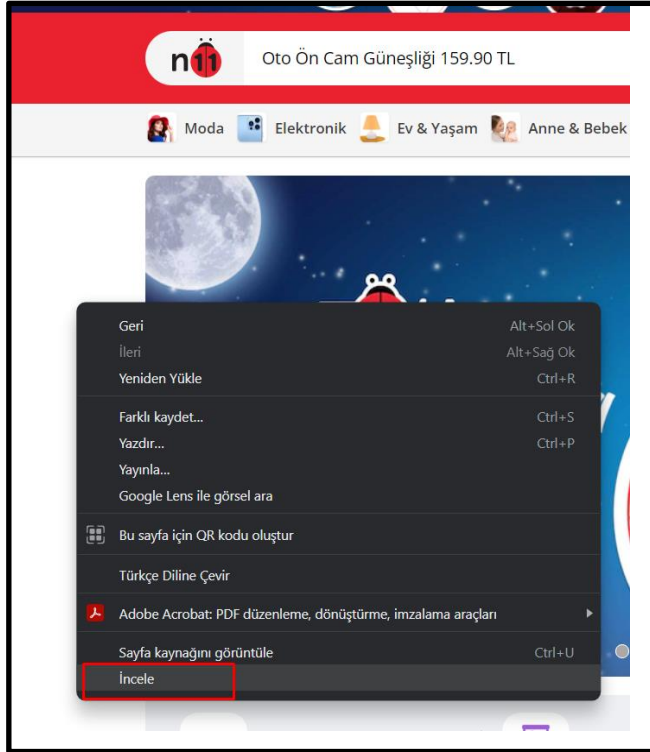
Syntax:



<start tag> content </end tag>



Inspect Page



Otomasyon için unique (tek) web elementlerini tanımlamak üzere HTML kodunu inceleyeceğiz.

Bunun için web sayfasında;

- ✓ Sağ Click
- ✓ İncele - Inspect

Inspect Page



```
▼ <div class="myAccountHolder customMenuItem withLocalization" style>
  ▼ <div class="myAccountNotLogin">
    ::before
    ▼ <div class="myAccount">
      <a href="https://www.n11.com/hesabim" rel="nofollow" title="HESABIM"
        class="menuTitle nobg">HESABIM</a>
      ▼ <div class="loginStatus clearfix"> flex
        ::before
        <a href="https://www.n11.com/uye-ol" class="btnSignUp" title="Üye Ol"
          rel="nofollow">Üye Ol</a> == $0
        ▼ <a href="https://www.n11.com/giris-yap" class="btnSignIn" title="Giriş
          Yap" rel="nofollow" refererrpolicy="no-referrer-when-downgrade">
            ::before
            "Giriş Yap"
          </a>
          ::after
        </div>
      </div>
    </div>
  </div>
```

There are different type of WebElement tags like;

<html>,
<body>,
<form>,
<label>,
<input>,
<div>,
<a>

Web elementleri birlikte kullanıcı ara yüzünde (UI) bir web sayfası oluştururlar.

WebElements Commands



sendKeys() command
isDisplayed() command
isSelected() command
isEnabled() command
submit() command
click() command
getLocation() command
clear() command
getText() command
getAttribute() command
getSize() command

Selenium WebElement Methods

```
WebElement element = driver.findElement(By.id("value"));  
element.  
clear() : void - WebElement  
click() : void - WebElement  
equals(Object obj) : boolean - Object  
findElement(By by) : WebElement - WebElement  
findElements(By by) : List<WebElement> - WebElement  
getAttribute(String name) : String - WebElement  
getClass() : Class<?> - Object  
getCssValue(String propertyName) : String - WebElement  
getLocation() : Point - WebElement  
getScreenshotAs(OutputType<X> target) : X - TakesScreenshot  
getSize() : Dimension - WebElement  
getTagName() : String - WebElement  
getText() : String - WebElement  
hashCode() : int - Object  
isDisplayed() : boolean - WebElement  
isEnabled() : boolean - WebElement  
isSelected() : boolean - WebElement  
notify() : void - Object  
notifyAll() : void - Object  
sendKeys(CharSequence... keysToSend) : void - WebElement  
submit() : void - WebElement  
toString() : String - Object  
wait() : void - Object  
wait(long timeout) : void - Object  
wait(long timeout, int nanos) : void - Object
```

WebElements Commands



sendKeys(): sendKeys command allows the user to type content automatically into an editable field while executing tests. It works with text entry elements such as INPUT and TEXTAREA
/It returns nothing

Syntax: `element.sendKeys("text");`
`element.sendKeys("text" + Keys.ENTER);`

Parametre olarak yazılan metni
Web Element'e gönderir.
İstediğimiz metni yollayıp, sonra
ENTER'a basar.

```
// Create WebElement
WebElement elesendKeys = driver.findElement(By.id("TextBox"));
// Perform sendKeys operation
elesendKeys.sendKeys("Cheese");
// OR
// Send value to particular WebElement e.g: Textbox.
driver.findElement(By.id("TextBox")).sendKeys("Cheese");
```

WebElements Commands



isDisplayed(): If a certain element is present and displayed / if the element is displayed, then the value returned is true. If not, then the value returned is a NoSuchElementException

Syntax: element.isDisplayed();

Web Element görünür ise true yoksa false döner.

```
WebElement element = driver.findElement(By.id("UserName"));
boolean status = element.isDisplayed();
//Or can be written as
boolean status = driver.findElement(By.id("UserName")).isDisplayed();
```

WebElements Commands



isSelected(): It is used to determine if an element is selected /If the specified element is selected, the value returned is true. If not, the value returned is false

Syntax: element.isSelected();

Web Element seçili ise true yoksa false döner.

```
WebElement element = driver.findElement(By.id("Sex-Male"));
boolean status = element.isSelected();
//Or can be written as
boolean staus = driver.findElement(By.id("Sex-Male")).isSelected();
```


WebElements Commands



isEnabled(): It verifies if an element is enabled on the web page

Syntax: `element.isEnabled();`

Web Element erişilebilir ise true yoksa false döner.

```
// Create WebElement
WebElement eleEnabled = driver.findElement(By.id("TextBox"));
// Perform isEnabled operation
eleEnabled.isEnabled();
// OR
// Verify WebElement is Enabled or Not? e.g: Radio / Checkbox.
driver.findElement(By.id("Text")).isEnabled();
```

WebElements Commands



submit(): It comes to interacting with forms on a web page, it doesn't require a parameter and returns nothing

Syntax: `element.submit();`

Web Element ile işlem yaparken ENTER tuşuna basma görevi yapar.

```
WebElement element = driver.findElement(By.id("SubmitButton"));
element.submit();
//Or can be written as
driver.findElement(By.id("SubmitButton")).submit();
WebElement element = driver.findElement(By.id("SubmitButton"));
element.submit();
//Or can be written as
driver.findElement(By.id("SubmitButton")).submit();
```

WebElements Commands



click(): Clicks a link, button, checkbox, or radio button

Syntax: `element.click();`

Web Element'e click yapar.

```
// Create WebElement
WebElement elecclick = driver.findElement(By.id("TextBox"));
// Perform click operation
elecclick.click();
// OR
// Click on any WebElement e.g: Button.
driver.findElement(By.id("Button_Id")).click();
// Create WebElement
WebElement elecclick = driver.findElement(By.id("TextBox"));
// Perform click operation
elecclick.click();
// OR
// Click on any WebElement e.g: Button.
driver.findElement(By.id("Button_Id")).click();
```

WebElements Commands



getLocation(): Retrieves the location of a specified element on a web page. It does not require a parameter and returns the Point object as its result. The X and Y coordinates of the element can be derived from the Point object returned.

Syntax: `element.getLocation();`

Web Element'e nokta olarak bakıldığında sol üst köşesiyle x ve y koordinatlarını döndürür.

```
WebElement element = driver.findElement(By.id("SubmitButton"));
Point point = element.getLocation();
System.out.println("X cordinate : " + point.x + "Y cordinate: " + point.y);
```

WebElements Commands



clear(): If the element in question is a text entry, its value will be cleared. It doesn't require a parameter and returns nothing.

Syntax: `element.clear();`

Metin alanındaki bir değeri
temizleme veya silme işlemidir.

```
// Create WebElement
WebElement eleClear = driver.findElement(By.id("TextBox"));
// Perform clear operation
eleClear.clear();
// OR
// Clear particular WebElement e.g: Textbox.
driver.findElement(By.id("TextBox")).clear();
```

WebElements Commands



getText(): This includes the inner text as well as the sub-elements sans whitespace. It doesn't require a parameter and returns a string value as its result. This method is often used to verify labels, messages, error, and other elements.

Syntax: `element.getText();`

Belirli bir Web Element içindeki metni alır.
Parametre gerektirmez ve sonuç olarak bir String değeri döndürür.

Bu yöntem genellikle web sitesi ziyaretçilerine gösterilen etiketleri, mesajları, hataları ve diğer öğeleri (metin içeren) doğrulamak için kullanılır.

```
// Create WebElement
WebElement elegetText = driver.findElement(By.id("TextBox"));
// Perform getText operation
elegetText.getText();
// OR
// Get text of Particular WebElement & Store into String
driver.findElement(By.id("TextBox")).getText();
```

WebElements Commands



getAttribute(): This command retrieves the attribute value of a specified element. It uses String as the parameter and returns a string value as its result.

Syntax: `element.getAttribute();`

Belirtilen bir Web Element'in attribute değerini alır. Parametre olarak String kullanır ve sonucu olarak bir String değeri döndürür.

```
WebElement element = driver.findElement(By.id("SubmitButton"));
String attValue = element.getAttribute("id"); //This will return "SubmitButton"
```

WebElements Commands



getSize(): Returns the width and height of a rendered element.

Syntax: `element.getSize();`

Belirli bir işlenmiş Web Element'in yüksekliğini ve genişliğini alır. Parametre gerektirmez ve sonucu olarak Dimension nesnesini döndürür.

```
WebElement element = driver.findElement(By.id("SubmitButton"));
Dimension dimensions = element.getSize();
System.out.println("Height :" + dimensions.height + "Width : " + dimensions.width);
```




THANKS!

Any questions?

Elly D. - Full-Stack Automation Engineer

Garry T. - Full-Stack Automation Engineer

