

SELENIUM 4

(YENILIKLER VE DEGISIKLIKLER)

Selenium 4 ile kullanabileceginiz yeni ozellikleri gunluk olarak sizlerle paylasacagim

Selenim 4 ve Selenium 3 karsilastirmasi

1) driver.manage().window().minimize()

Se Selenium 3

```
driver.manage().window().
```

- (m) maximize()
inst expr inst
- (m) fullscreen()
- (m) getPosition()
- (m) getSize()
- (m) setPosition(Point ta
- (m) setSize(Dimensi
- (m) equals(Object o

Se Selenium 4

```
driver.manage().window().
```

- (m) maximize()
inst expr instance
- (m) fullscreen()
- (m) getPosition()
- (m) getSize()
- (m) minimize()
- (m) setPosition(Point ta
- (m) setSize(Dimension ta

Selenium 3'de var olan window manage ozelliklerine ek olarak **minimize()** eklendi.
minimize() ile normal hayatta yaptiginiz sekilde window'u sembol olarak asagi indirebilirsiniz.

2) driver.switchTo ().newWindow();



```
driver.switchTo().  
  (m) window(String nameOrId)  
  (m) frame(WebElement frame)  
  (m) frame(int index)  
  (m) alert()  
  (m) frame(String nameOrId)  
  (m) defaultContent()  
  (m) activeElement()  
  (m) parentFrame()  
  (m) equals(Object obj)
```



```
driver.switchTo().  
  (m) newWindow(WindowType)  
  (m) window(String nameOrId)  
  (m) frame(WebElement frame)  
  (m) alert()  
  (m) frame(int index)  
  (m) frame(String nameOrId)  
  (m) defaultContent()  
  (m) activeElement()  
  (m) parentFrame()  
  (m) equals(Object obj)
```

Selenium 3'de var olan switchTo () ozelliklerine ek olarak newWindow() eklendi. Yeni pencereyi isterseniz TAB, isterseniz yeni WINDOW olarak acabilirsiniz.

3) webElement.get



```
webElement.get|
  (m) getText()
  (m) getAttribute(String name)
  (m) getCssValue(String propertyName)
  (m) getLocation()
  (m) getRect()
  (m) getSize()
  (m) getTagName()
```



```
webElement.get|
  (m) getText()
  (m) getAttribute(String name)
  (m) getSize()
  (m) getAccessibleName()
  (m) getAriaRole()
  (m) getCssValue(String propertyName)
  (m) getDomAttribute(String name)
  (m) getDomProperty(String name)
  (m) getLocation()
  (m) getRect()
  (m) getShadowRoot()
  (m) getTagName()
```

Selenium 3'de Web Element'ler için var olan Size, Text, Location ve Rect method' larına ilave olarak

Web Element'in AccessibleName ve Aria role özelliklerini öğrenebileceğimiz method'lar eklendi

```
=====Selenium 3 de olanlar=====
Size : (344, 38)
Text :
Location : (267, 11)
Rect.Height : 38
=====Selenium 4 de yenilikler=====
AccessibleName : Search
Aria role : textbox
```

4) webElement.get



```
webElement.get|
  (m) getText()
  (m) getAttribute(String name)
  (m) getCssValue(String propertyName)
  (m) getLocation()
  (m) getRect()
  (m) getSize()
  (m) getTagName()
```

Selenium 3'de Web Element'ler için var olan istenen attribute'un value'su, web elementin tagName'i ve spesifik CSS özelliklerinin yanına

Web Element'in Dom ile erişebilmeniz için attribute'lerin Dom değeri ve Dom property değerini öğrenebileceğiniz method'lar eklendi



```
webElement.get|
  (m) getText()
  (m) getAttribute(String name)
  (m) getSize()
  (m) getAccessibleName()
  (m) getAriaRole()
  (m) getCssValue(String propertyName)
  (m) getDomAttribute(String name)
  (m) getDomProperty(String name)
  (m) getLocation()
  (m) getRect()
  (m) getShadowRoot()
  (m) getTagName()
```

```
=====Selenium 3 de olanlar=====
Id attribute'nun değeri : twotabsearchtextbox
webElement'in Css background color değeri : rgba(0, 0, 0, 0)
webElement Tag name'i : input
=====Selenium 4 de yenilikler=====
Id attribute'nun Dom attribute değeri : twotabsearchtextbox
Id attribute'nun Dom property değeri : null
```

5) Relative Locators



```
driver.findElement(By)
```

- By org.openqa.selenium
- By.id(String id) (org.op
- By.tagName(String tagName
- By.xpath(String xpathExp
- By.linkText(String linkT
- By.className(String clas
- By.name(String name) (or
- By.partialLinkText(Strir
- By.cssSelector(String cs



```
driver.findElement(with(By))
```

- By org.openqa.selenium
- By.id(String id) (org.open
- By.tagName(String tagName)
- By.linkText(String linkTex
- By.xpath(String xpathExpre
- By.className(String classN
- By.name(String name) (org.
- By.partialLinkText(String
- By.cssSelector(String cssS

Selenium 3'de Web Element'leri locate etmek için kullandığımız 8 adet locator Selenium 4'de de var

Var olan bu method'ların yanına Selenium 4

Devrim gibi bir özellik eklemiş

Eklenen RelativeLocator Class'ındaki method'lar sayesinde bir web elementi etrafındaki diğer web elementler ile tanımlayabilirsiniz

```
driver.findElement(RelativeLocator.with(By.id("x")).);
```

- toRightOf(WebElement)
- below(WebElement)
- above(WebElement)
- toRightOf(By locator)
- below(By locator)
- above(By locator)
- near(By locator)
- near(WebElement)
- near(By locator)
- near(WebElement)
- toLeftOf(By locator)
- toLeftOf(WebElement)
- findElements(Selector)

6) Relative Locators ornek

Bazi web elementlerde developer'lar dinamik ozellikler kullandiklarindan sabit bir locator bulamayabilirsiniz.

Bu durumda Selenium 4 imdadınıza yetisip, gunluk hayatimizda kullandigimiz adres tariflerine benzer bir imkan sunuyor.

Once **with** ile var olan her hangi bir ozelligini soyleyip sonra etrafındaki web elementleriyle tarif verebilirsiniz

Asagidaki ornekte **boston** ve **sailor** web elementlerini locate edip, onlar üzerinden **berlin** ve **mountie** web elementlerini locate ettim.



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


7) manage().timeouts()



```
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  
driver.manage().timeouts().  
    implicitlyWait(long l, TimeUnit tim... Timeo  
    pageLoadTimeout(long l, TimeUnit ti... Timeo  
    inst expr instanceof Type ? ((Type) expr).  
    setScriptTimeout(long l, TimeUnit t... Timeo
```



Selenium 3'de wait için zaman tanımlarken kullandığımız long TimeUnit kaldırılmış,
Yerine java util kutuphanesinden Duration class'ından kullanacağız
ofSeconds(sure) metotları getirilmiş.

```
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));  
driver.manage().timeouts().  
    implicitlyWait(Duration duration) Tim  
    implicitlyWait(long time, TimeUnit... Tim  
    inst expr instanceof Type ? ((Type) expr  
    scriptTimeout(Duration duration) Tim  
    getScriptTimeout() Dur  
    getImplicitWaitTimeout() Dur  
    getPageLoadTimeout() Dur  
    pageLoadTimeout(Duration duration) Tim
```


8) Explicitly wait



```
WebDriverWait wait=new WebDriverWait(driver, timeoutInSeconds: 20);
```



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

Selenium 3'de time icin kullanılan timeoutInSeconds Class'inin yerine Selenium 4'de Duration Class'i kullanilmaya basladi.

Bu degisim implicitly wait icin manage().timeouts()'da oldugu gibi explicitly wait'de wait objesi olustururken de karsimiza cikiyor.

9) WebDriverManager.

Se Selenium 3

```
WebDriverManager.  
  m chromedriver()  
  m edgedriver()  
  m operadriver()  
  m firefoxdriver()  
  new  
  m chromiumdriver()  
  m getInstance(Class<?> webDri  
  m getInstance(DriverManagerTy  
  m globalConfig()  
  m iedriver()  
  m main(String[] args)  
  m phantomjs()  
  m seleniumServerStandalone()
```

Se Selenium 4

```
WebDriverManager.  
  m chromedriver()  
  m operadriver()  
  m edgedriver()  
  m firefoxdriver()  
  new  
  m chromiumdriver()  
  m getInstance()  
  m getInstance(String b  
  m getInstance(Class<?>  
  m getInstance(DriverMa  
  m iedriver()  
  m main(String[] args)  
  m safaridriver()
```

Selenium 3'de desteklenen browser'lardan phantomjs Selenium 4'de kaldırılırken, Selenium 3'de direk kullanılmayıp getInstance () ile kullanılabilen safaridriver selenium 4'de direk kullanıma acılmış

NOT : Bazi kaynaklarda Selenium 4'de Opera desteginin de kaldırıldığı yazsa da yukarıda gorebileceğiniz sekilde operadriver listede yerini alıyor

10) ScreenShot



```
public static String getScreenshot(String name) throws IOException {
    // naming the screenshot with the current date to avoid duplication
    String date = new SimpleDateFormat( pattern: "yyyyMMddhhmmss").format(new Date());

    // TakesScreenshot is an interface of selenium that takes the screenshot
    TakesScreenshot ts = (TakesScreenshot) Driver.getDriver();
    File source = ts.getScreenshotAs(OutputType.FILE);
    // full path to the screenshot location
    String target = System.getProperty("user.dir") + "/test-output/Screenshots/" + name + date + ".png";

    File finalDestination = new File(target);
    // save the screenshot to the path given
    FileUtils.copyFile(source, finalDestination);

    return target;
}
```



```
@Test
public void test01() throws InterruptedException, IOException {
    WebDriverManager.chromedriver().setup();
    driver = new ChromeDriver();

    driver.get("https://www.amazon.com");
    Thread.sleep( millis: 3000);

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

    File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
    FileUtils.copyFile(scrFile, new File( pathname: "./tumSayfa.png"));

    File scrFile1 = aramaKutusu.getScreenshotAs(OutputType.FILE);
    FileUtils.copyFile(scrFile1, new File( pathname: "./aramaKutusu.png"));
}
```

Selenium 3 de ekran fotografi almak mumkundu ancak bir method olusturmak ve onu kullanmak gerekiyordu, soldaki ornekte onceki framework'de kullandigim, sadece tum sayfanin görüntüsünü kaydeden metodu gorebilirsiniz.

Selenium 4'de tum ekranin fotografini veya istediginiz bir WebElement'in görüntüsünü kolaylikla alabilirsiniz. Sagdaki ornekte Amazon sayfasina gidip hem tum sayfanin hem de arama kutusunun görüntüsünü 2 satir kod yazarak aldım