



Selenium

Chapter 01



Neler Öğreneceğiz?



- Software Testing
- Manuel & Automation Testing
- Selenium Avantaj ve Dezavantajları Nelerdir?
- Test Automation Framework
- Selenium Project
- Selenium WebDriver - Commands
 - Browser Commands
 - Navigation Commands
 - Driver Manage Windows Commands
- Thread Sleep
- Testing On Firefox Browser



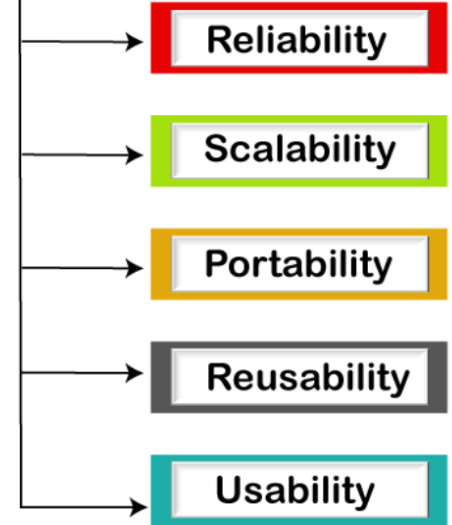
Software Testing



Yazılım testi, yazılımın tüm niteliklerini (Güvenilirlik, Ölçeklenebilirlik, Taşınabilirlik, Yeniden Kullanılabilirlik, Kullanılabilirlik) göz önünde bulundurarak yazılımın doğruluğunu belirleme ve yazılım hatalarını veya hatalarını veya kusurlarını bulmak için yazılım bileşenlerinin yürütülmesini değerlendirme sürecidir.

Test yapmak zorunludur, çünkü yazılımın test eksikliği nedeniyle herhangi bir zamanda başarısız olması tehlikeli bir durum olacaktır. Bu nedenle, test yazılımı olmadan son kullanıcıya dağıtılamaz.

Attributes of Software



Software Testing Önemi



- Yazılımı geliştirirken programcı tarafından oluşturulabilecek hataları bulmak.
- Kalite düzeyi konusunda güven sağlamak ve bilgi vermek.
- Kusurları önlemek için.
- Nihai sonucun iş ve kullanıcı gereksinimlerini karşıladığından emin olmak için.
- İş Gereksinimi ve sistem gereksinimini karşıladığından emin olmak için.
- Kaliteli bir ürün sunarak müşterilerin güvenini kazanmak.



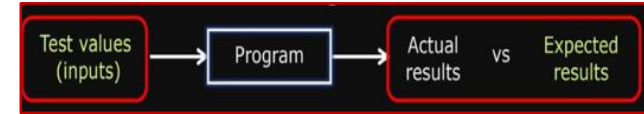
Software Testing



EXPECTED RESULT (beklenen sonucun), **ACTUAL RESULT** (gerçek sonuca) eşit olup olmadığını kontrol etme işlemidir.

Eğer Expected Result = Actual Result, ise status **PASSED** → Happy Path

Eğer Expected Result != Actual Result, ise status **FAIL**



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

Her User Story için **Positif** ve **Negatif Test**(ler) yapılmalıdır

Test müşteri/işletme ihtiyaçlarını karşılamak için yapılır.

Bir uygulamayı test etmek için önceden belirlenmiş user storyler (kullanıcı hikayeleri) ve tanımlanmış acceptance criterias (kabul kriterleri) dikkate alınır.

Happy Path: Sorun oluşturmayacağı bilinen girdilerle test etme yöntemidir. **Golden Path** ve **Happy Day** scenario olarak da bilinmektedir.

Örneğin verilen kredi kartı numarasının geçerliliğini doğrulayan bir fonksiyonumuz var diyelim. Bu fonksiyon için mutlu yol, fonksiyonun sonuna kadar işletilebildiği, onaylama kurallarının hiçbirinin hata üretmediği ve fonksiyonun pozitif sonuç verdiği bütün senaryolara denilebilir.

User Story – Acceptance Criteria



- **User Story:** Kullanıcı uygulamada oturum açabilmelidir.

- **Acceptance Criteria:**

Given kullanıcı giriş sayfasına gider.

When Kullanıcı kimlik bilgilerini girer.

Then Kullanıcı hatasız varsayılan sayfada olmalıdır.

DEVELOPER → Dev. environment'da geliştirir, test ortamında kodu dağıtır.

TESTER -> Test ortamında test başlatır ve test datalarını (URL, ortam, kimlik bilgileri, kullanıcı adı, şifre) hazırlar. Aynı zamanda test caseleri oluşturur.



Test Case



- TC01_kullanıcı doğru kimlik bilgilerini kullanarak oturum açabilmelidir.

(POZİTİF TEST ZORUNLUDUR)

Doğru kullanıcı adı ve doğru şifre ile test edilir.

- TC02_kullanıcı yanlış kimlik bilgileriyle oturum açamamalıdır.

**(İSTEĞE BAĞLIDIR ANCAK YENİ BİR İŞLEVSELLİK OLUŞTURULDUYSA
NEGATİF OLARAK SENARYO KULLANILMALIDIR)**

Boş kullanıcı adı veya boş şifre.

Kullanıcı adı doğru ama şifre yanlış.

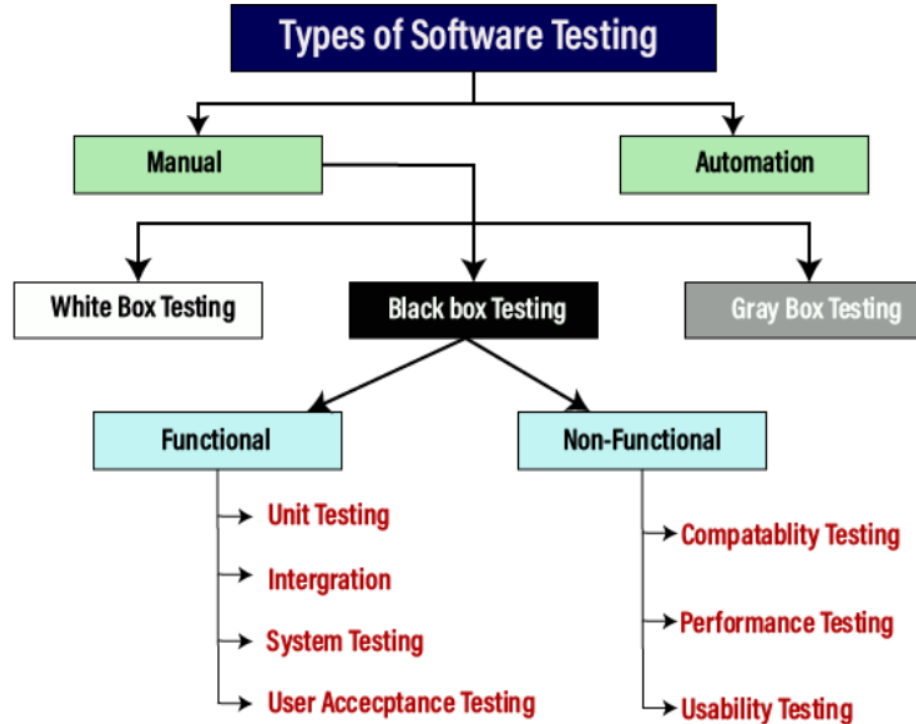
Doğru şifre ama yanlış kullanıcı adı.

Bu tarz senaryolar ile test edilir.

Test Çeşitleri (Testing Types)



Bir uygulamanın işlevselliğini, herhangi bir otomasyon aracı yardımı almadan müşterinin ihtiyaçlarına göre kontrol etme sürecidir.



Herhangi bir manuel test senaryosunu, otomasyon araçları yardımıyla test komut dosyalarına dönüştürme işlemidir.

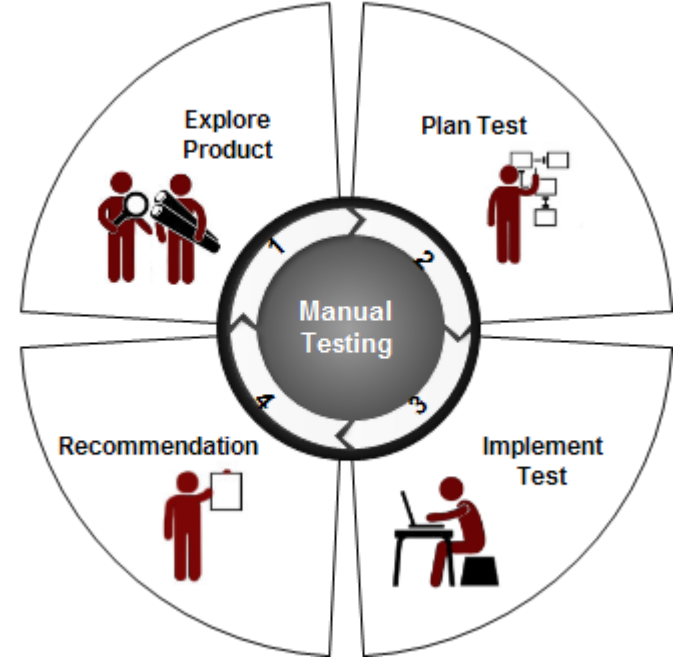
Manual & Automation Testing



Manual Testing



- Manuel test, bir otomasyon test aracı yardımı almadan yapılan test türüdür.
- Geliştirilmekte olan yazılımdaki hataları bulmak için yapılır.
- Manuel testte, bir kişi testleri adım adım test komut dosyaları olmadan gerçekleştirir.
- Tüm test türleri arasında klasik bir yöntemdir. Bu süreçte yazılım test uzmanları, farklı modüller için test senaryoları hazırlar, test senaryolarını yürütür ve çalıştırır.
- En kısa sürede test etmeniz gereken bir şey varsa ve bu sadece tek seferlik bir işse, sizin için en mantıklı şey manuel test yapmak olabilir.
- Manuel test çok pratiktir. Analistlerin ve kalite kontrol mühendislerinin kolayca test yapmasını sağlar.



Manual Testing Avantajları



- Otomasyon araçları için bütçenizi harcamanıza gerek olmadığından daha ucuzdur.
- Küçük bir değişikliği test ederken, bir otomasyon testi zaman alıcı olabilecek kodlama gerektirecektir. Siz anında manuel olarak test edebilirsiniz.
- Manuel test programlama bilgisi olmadan mümkündür.
- Manuel test karmaşık senaryoları daha iyi idare eder.

Manual Testing Dezavantajları

- Bir insan tarafından yürütüldüğü için daha az güvenilir bir test yöntemidir. Bu nedenle, her zaman hatalara eğilimlidir.
- Manuel testi tekrar kullanmak mümkün değildir.
- Manuel test, bir test uzmanı için zaman alan bir süreçtir.
- Manuel test yavaş ve zahmetlidir.

Automation Testing



- Otomasyon testi yani diğer bir adıyla otomatik test, test uzmanlarının test yürütmeyi otomatikleştirmek için kod yazarak test komut dosyaları oluşturduğu bir süreçtir.
- Otomasyon test uzmanları, bu testleri çalıştırmak için otomasyon araçlarını kullanır.
- Test otomasyonunun amacı, testi daha kısa sürede ,daha iyi ve daha güvenilir yollarla gerçekleştirmektir.
- Otomatik test, tamamen otomatik olarak çalışan bir test türüdür.
- Test uzmanının bir uygulamanın beklendiği gibi çalışıp çalışmadığını belirlemesine yardımcı olur.
- Otomasyon testi, insan hatası içermediği için daha doğru sonuç verir.
- Eğer büyük bir projeniz varsa otomasyon testi sizin için çok kullanışlı olabilir.

Automated Testing



Automation Testing Avantajları

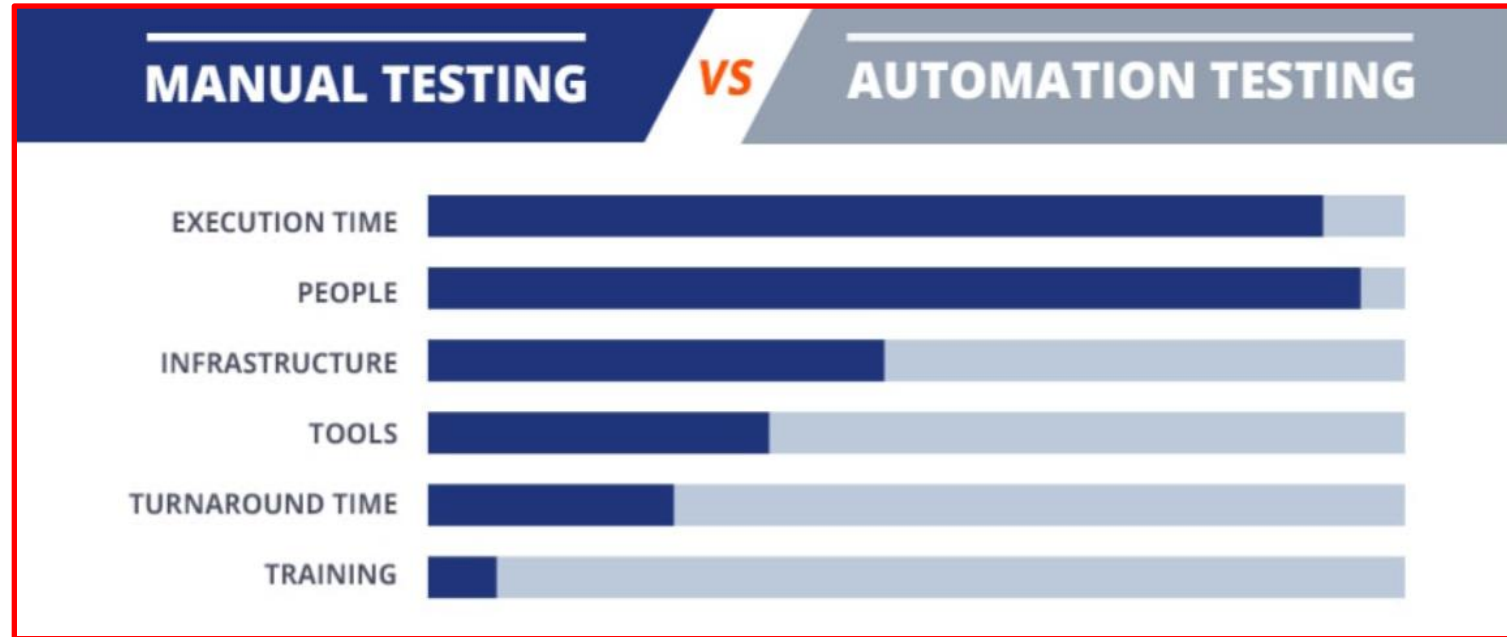


- Otomasyon süreci kod tabanlı olduğu için güvenilirdir.
- Otomasyon testi çok hızlı ve verimlidir.
- Otomatik test, daha fazla hata bulmanıza yardımcı olur.
- Otomasyon süreci kayıt altına alınabilir. Böylece, yeniden kullanmanıza olanak tanır.

Automation Testing Dezavantajları

- Otomasyon testi programlama bilgisi gerektirir.
- Otomasyon testini yürütmek için kullanılan araçlar pahalı olabilir ve bu da test projesinin maliyetini artırabilir.
- Otomatik test, kodlama ve test bakımı gerektirir ve test bakımı maliyetlidir.

Manual vs. Automation Test



Manual & Automation Test Farkları



- Manuel test, QA analisti (insan) tarafından manuel olarak yapılırken, otomasyon testi, bir test cihazı tarafından komut dosyası, kod ve otomasyon araçları (bilgisayar) kullanılarak yapılır.
- Manuel test süreci, insan hataları nedeniyle doğru değildir, oysa otomasyon süreci kod ve komut dosyası tabanlı olduğu için güvenilirdir.
- Manuel test, zaman alan bir süreçtir, otomasyon testi ise çok hızlıdır.
- Manuel test programlama bilgisi olmadan mümkündür, oysa otomasyon testi programlama bilgisi olmadan mümkün değildir.

Selenium



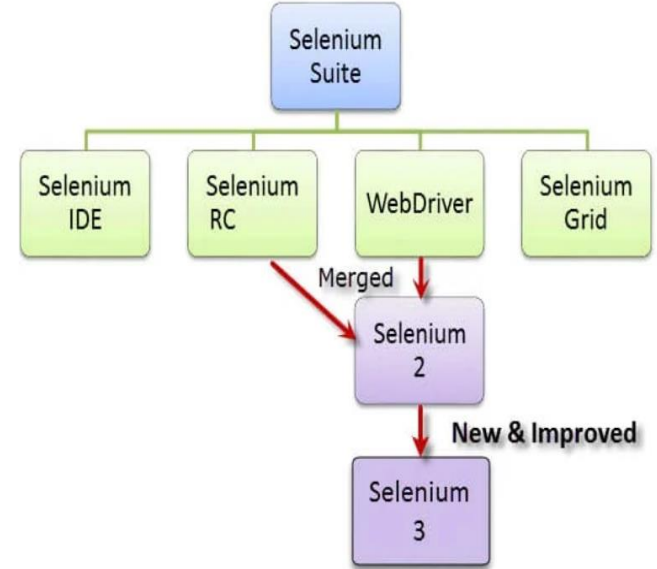
- Farklı tarayıcılarda yada ortamda web uygulamalarını test etmek için kullanılan test çerçevesidir. (framework)
- Başka bir deyişle, **web uygulamalarının** kalite güvencesi için kullanılan yazılım test otomasyon araçlar paketidir.
- Ayrıca açık kaynaklı yazılım olduğu için ücretsizdir.
- Test Komut Dosyaları için C#, Python, Java gibi birden fazla programlama dili kullanılır.
- Selenium Testi, Selenium test araçlarını kullanılarak yapılan testlerin genel ismidir.
- 2021'de piyasaya sürülen Selenium 4'ü öğreneceğiz.
- Selenium, web arayüz testleri yapmak için (GUI test), web tarayıcılarına komut göndermeye yarayan bir **KÜTÜPHANEDİR**.
- Selenium bir programlama dili, bir script dili veya masaüstü bir uygulama değildir.
- Selenium, otomasyon yapmak için kendi sınıflarına ve yöntemlerine sahip bir suite'dir.

Selenium Components (Bileşenleri)

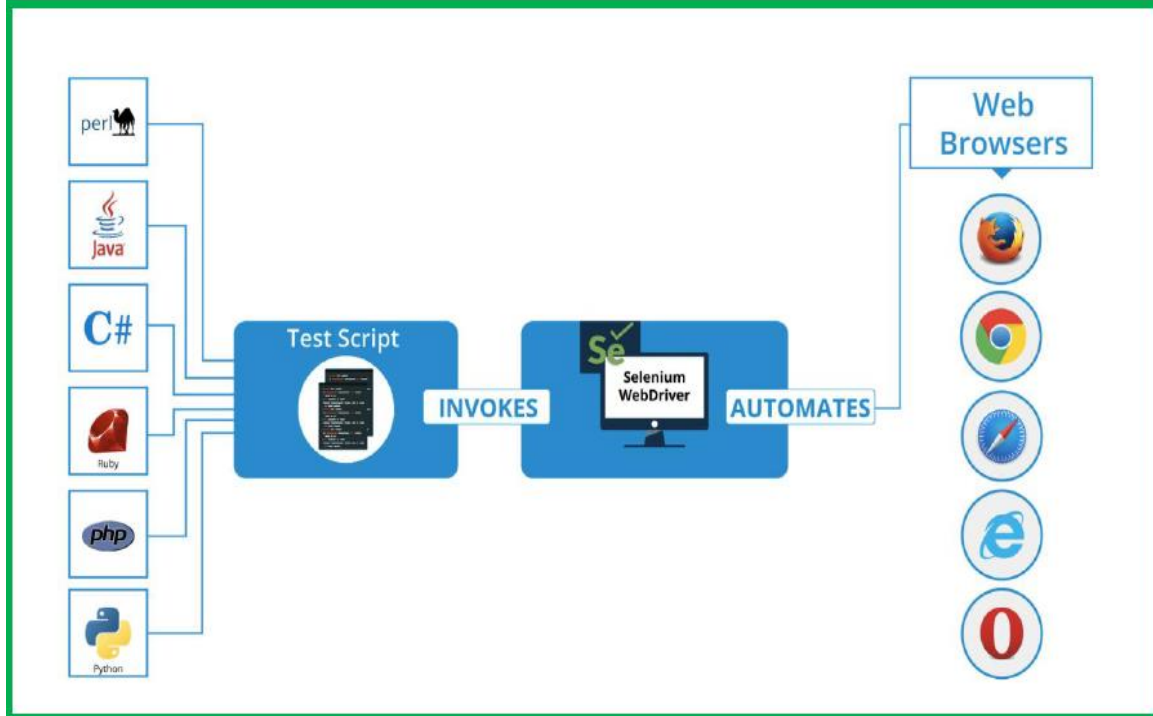


Selenium sadece bir araçtan oluşmamaktadır ve aşağıdaki gibi 3 farklı bileşeni vardır;

- Selenium IDE (Integrated Development Environment)
- **Selenium WebDriver**
- Selenium Grid



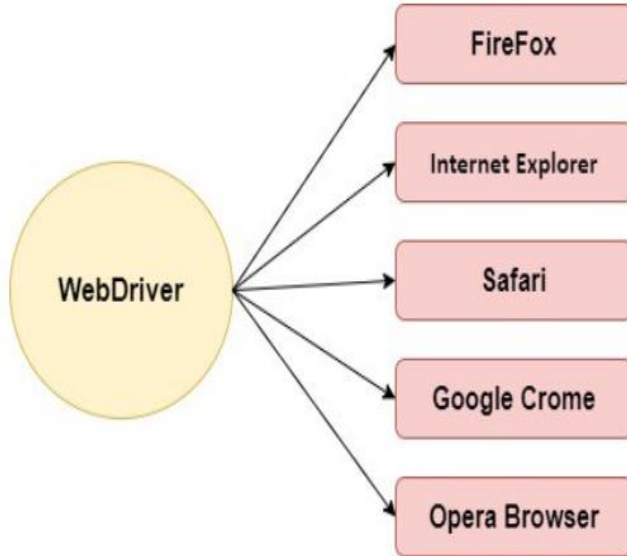
Selenium Nasıl Çalışır?



- 1) Driver Selenium'dan komutları alır.
- 2) Bu komutları tarayıcının API'nda dönüştürür.
- 3) Browser'lardan dönen sonuçları alır ve Selenium'a geri gönderir.



Selenium Avantajları



1) Ücretsiz ve açık kaynaklıdır.

(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

Selenium Dezavantajları



- Masaüstü uygulamaları için otomasyon testini desteklemez.
- Açık kaynaklı bir yazılım olduğundan, teknik sorunlarınızı çözmek için topluluk forumlarına güvenmeniz gerekir.
- Selenium WebDriver'da test betikleri (script) oluşturmak için desteklenen programlama dillerinden en az birini bilmeliyiz.
- Selenium'un herhangi bir dahili raporlama özelliği yoktur; test raporları için **JUnit** veya **TestNG** gibi eklentilere güvenmeniz gerekir.
- Mobil test yapmak için üçüncü taraf (third party) eklentilere ihtiyacımız vardır.



Selenium & Chrome Driver Kurulumu



- 1) <https://www.selenium.dev/downloads/> adresine gidelim.
- 2) Selenium Client & WebDriver Language Bindings altında Java driver'ını yükleyelim.
- 3) Browsers altında Chrome Documentation linkini tıklayalım.
Chrome'un kendi sayfasına gidip Current Stable Release'i tıklayıp size uygun olanı indirelim.
*** Buradaki sürüm ile bilgisayarınızdaki Chrome sürümünün aynı olduğundan emin olun.
- 4) C:\Users\Garry\Documents altında **Selenium Dependencies** klasörü oluşturun.
- 5) Bu klasör altında **Drivers** ve **Libraries** klasörleri oluşturun.
- 6) İndirdiğimiz chromedriver'i **Drivers** klasörüne, selenium-java dosyasını ise **Libraries** klasörüne çıkartalım.
- 7) IntelliJ'de yeni project / package / class oluşturalım ve class içinde main method oluşturalım.
- 8) File/Project Structure/Modules/Dependencies kısmından jar dosyalarını yükleyelim.



Basic Testing on Chrome

TIME FOR TASK

1. Yeni bir class oluşturun: Class name \Rightarrow BasicNavigations
2. Create main method.



3. (macOS)

```
System.setProperty("", ""); System.setProperty("webdriver.chrome.driver", "/Users/Garry/Documents/seleniumdependencies/drivers/chromedriver");
```

3. (Windows)

```
System.setProperty("webdriver.chrome.driver", "C:\\Users\\Garry\\Documents\\seleniumdependencies\\drivers\\chromedriver.exe");
```

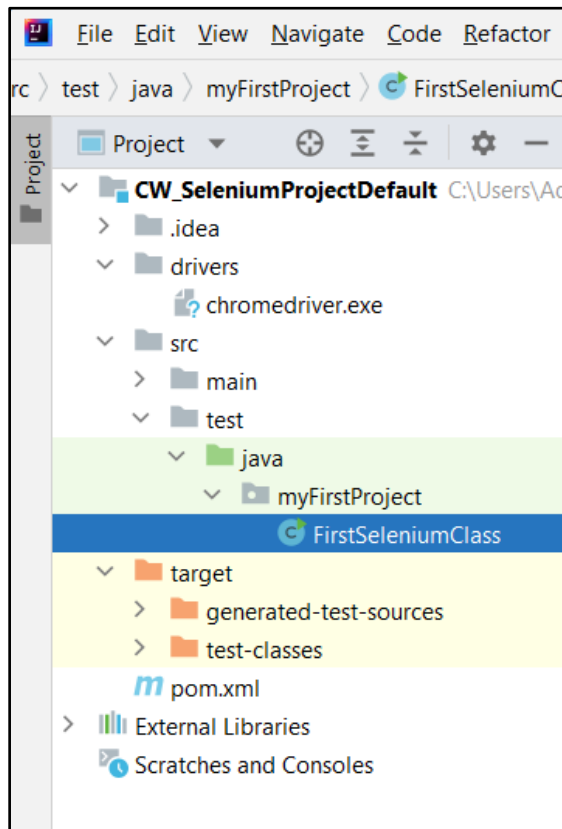
4. Chrome driver oluşturun.

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

5. "https://www.google.com" adresinden google ana sayfasını açın.

EXTRA

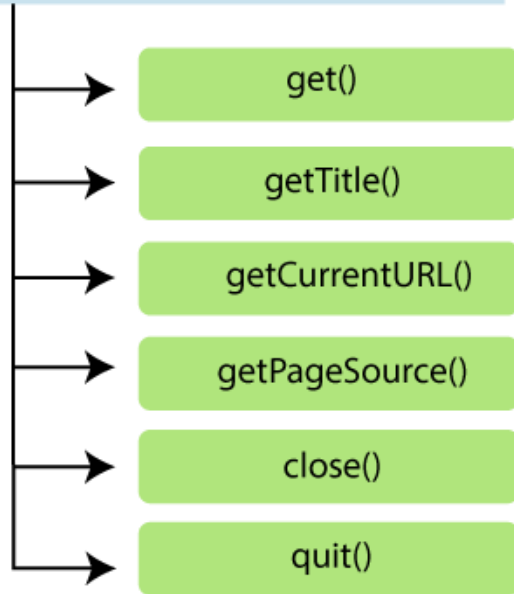
Test Automation Framework



Browser Commands



Browser Control Commands



Browser Commands



Get (String URL): Mevcut tarayıcı penceresine yeni bir web sayfası yükler. String'i parametre olarak kabul eder ve void değerini döndürür.

GetTitle(): Geçerli web sayfasının başlığını getirir. Hiçbir parametre kabul etmez ve bir String döndürür.

GetCurrentURL(): Geçerli web sayfasının Geçerli URL'sini temsil eden dizeyi getirir. Hiçbir şeyi parametre olarak kabul etmez ve bir String değeri döndürür.

Close(): Geçerli zamanda WebDriver tarafından çalışan geçerli tarayıcı penceresini sonlandırır. Geçerli pencere WebDriver tarafından çalışan tek pencere ise, tarayıcıyı da sonlandırır. Bu yöntem parametre olarak hiçbir şeyi kabul etmez ve void değerini döndürür.

Quit(): WebDriver tarafından çalışan tüm pencereleri sonlandırır. Tüm sekmeleri ve tarayıcının kendisini sonlandırır. Hiçbir şeyi parametre olarak kabul etmez ve void döndürür.

GetPageSoure(): Geçerli tarayıcıda yüklü olan geçerli web sayfasının kaynak kodunu döndürür. Hiçbir şeyi parametre olarak kabul etmez ve bir String değeri döndürür.

Browser Commands on Chrome



TIME FOR TASK

- Create main method
- Set Path
- Create chrome driver
- Open google home page: <https://www.google.com>
- Get Title on page
- Get Current URL on page
- Close/Quit the browser



Navigation Commands



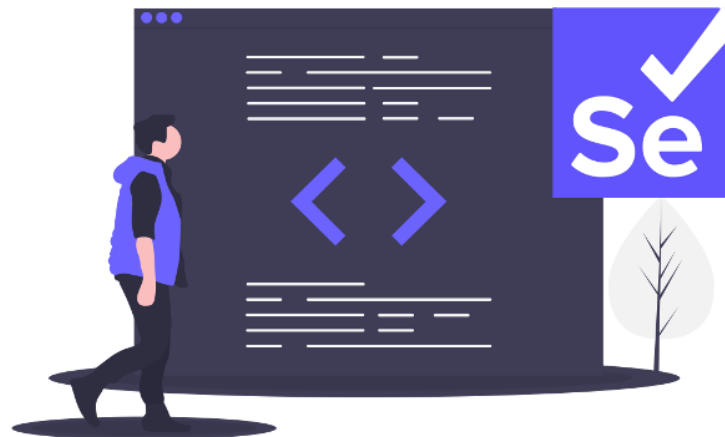
Navigation Control Commands

`back()`

`forword()`

`refersh()`

`to()`





Navigation Commands

driver.navigate().to (String URL): Bu yöntem, mevcut tarayıcı penceresine yeni bir web sayfası yükler. String'i parametre olarak kabul eder ve void değerini döndürür. get komutu (driver.get(URL) ile aynıdır.

driver.navigate().back (): Bu yöntem, web tarayıcısının mevcut tarayıcı penceresindeki **geri düğmesine** tıklamasını sağlar.

driver.navigate().forward (): Bu yöntem, web tarayıcısının mevcut tarayıcı penceresindeki **ileri düğmesine** tıklamasını sağlar.

driver.navigate().refresh (): Bu yöntem, mevcut web sayfasını mevcut tarayıcı penceresinde yeniler/yeniden yükler.



Navigation Commands on Chrome



Invoke Chrome Browser

Navigate to URL: <https://clarusway.com/>

Navigate to URL: <http://amazon.com/>

Come back to the Clarusway using the back command.

Again go back to the amazon website using forward command

Refresh the Browser using refresh command.

Close the Browser.



Driver Manage Window Commands



- `driver.manage().window().maximize()`: Browseri maximize yapar.
- `driver.manage().window().minimize()`: Browseri minimize yapar.
- `driver.manage().window().fullscreen()`: Browseri full screen yapar.
- `driver.manage().window().getPosition()`: Browserin koordinatlarını verir.
- `driver.manage().window().getSize()`: Browserin ölçülerini verir.





Thread.sleep()

```
Thread.sleep( millis: 5000 );
```

Geçerli Thread'in yürütülmesini birkaç saniyeliğine durdurmak için kullanılan Thread sınıfıyla ilgilidir.

Sleep() yöntemi, uyku süresini milisaniye cinsinden alır.

Sleep() yöntemini kullandığımızda ve yürütmeyi durduğu yerden yeniden başlattığımızda monitörün mülkiyeti kaybolmaz.

Basit bir deyişle Sleep() yöntemi, mevcut iş parçasını "Çalıştırılmaz" durumuna göndermekten sorumludur.

**Thread.sleep() execution'ı durdurur ve Java'dan gelir.
Kullanıldığı yerde exception fırlattırır.**

Driver Manage Window Commands on Chrome



- Invoke Chrome Browser
- Open URL: <https://www.google.com/>
- Maximize the window.
- Print the position and size of the page.
- Minimize the page.
- Wait 5 seconds in the icon state and maximize the page.
- Print the position and dimensions of the page in maximized state.
- Make the page fullscreen.
- Close the Browser.

Driver Manage Window Commands



- **`driver.manage().window().setPosition(new Point (x:80 , y:10))`** : Browseri istenen koordinata taşır.
- **`driver.manage().window().setSize(new Dimension(800 , 600))`**: Browseri istenen ölçülere getirir.
- **`driver.manage(). timeouts().implicitlyWait(Duration.ofSeconds(15))`**: İstenen sayfa açılıncaya kadar max. bekleme süresini belirtir.



Driver Manage Window Commands on Chrome



TIME FOR TASK

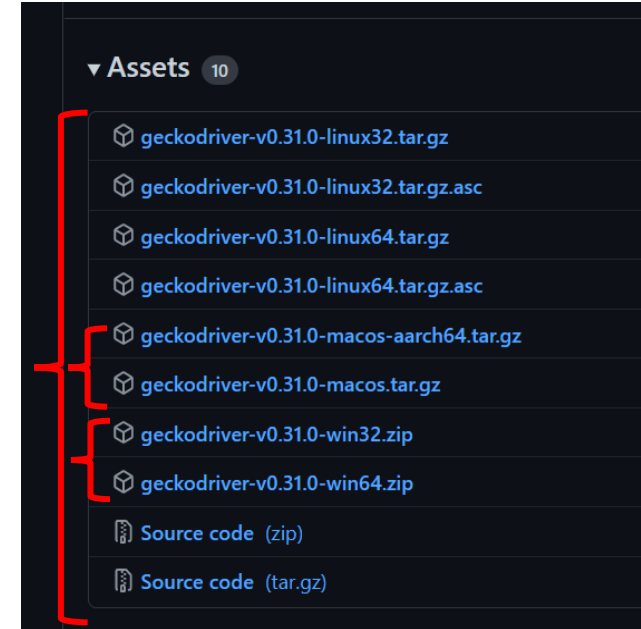
- Go to the Amazon URL : <https://www.amazon.com/>
- Print the position and size of the page.
- Adjust the position and size of the page as desired.
- Test that the page is in the position and size you want.
- Close the page.



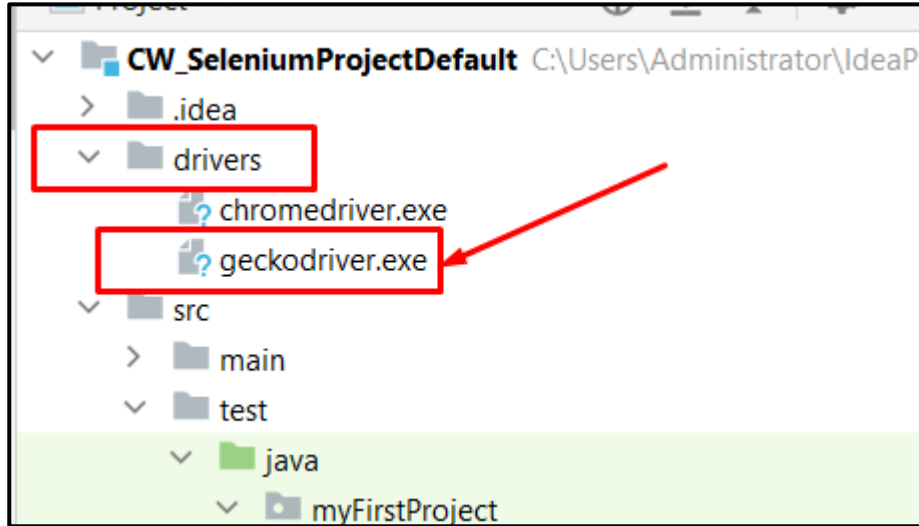
Testing on Firefox Browser



- <https://www.mozilla.org/en-US/firefox/new/>
Uygulama olarak kullanmak için
- <https://github.com/mozilla/geckodriver/releases>
geckodriver.exe için
for MacOS →
for Windows →



IDE'ye Nasıl Eklenir?



Class'da kullanabilmek için Chrome driver gibi **path** gösterilir.

WebDriver driver = new ChromeDriver();
Chrome driver oluşturmak için kullanılır.

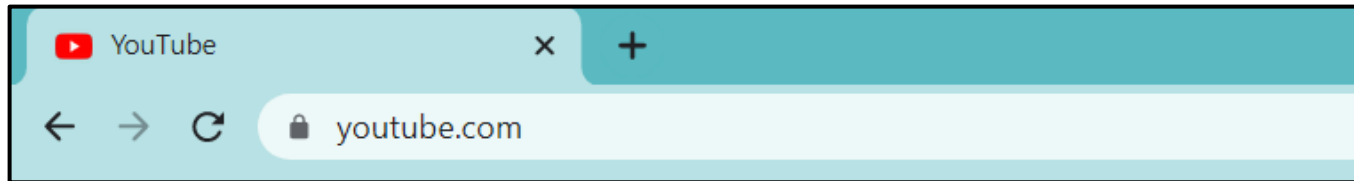
WebDriver driver = new FirefoxDriver();
Firefox driver oluşturmak için kullanılır.



Verify Title Test



- Invoke FireFox Driver
- Go to <https://www.youtube.com/>
- Verify the page title contains the word video.
- Close the driver.



Verify Title – URL Test



TIME FOR TASK

- Invoke Chrome Driver
- Navigate to Facebook homepage URL: <https://www.facebook.com/>
- Verify expected equals actual URL.
- Verify expected contains Facebook Word.



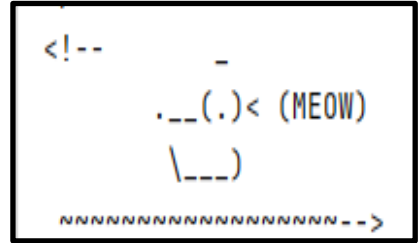
Page Source Test



- Go to the Amazon URL: <https://www.amazon.com/>
- Verify that it writes "Performance Metrics" and "MEOW" from the Source code.

NOT:

- `getPageSource()` sayfa kaynak kodunu bir String olarak döndürür.
- Sayfanın tamamında belirli bir metnin olup olmadığını doğrulamak için bunu kullanabiliriz.
- `pageSource`'un doğrulama yapmak için pek kullanılmadığını unutmayın.
- Çünkü çok geneldir ve yanlış sonuçlara yol açabilir.





THANKS!

Any questions?

Elly D. - Full-Stack Automation Engineer

Garry T. - Full-Stack Automation Engineer

