

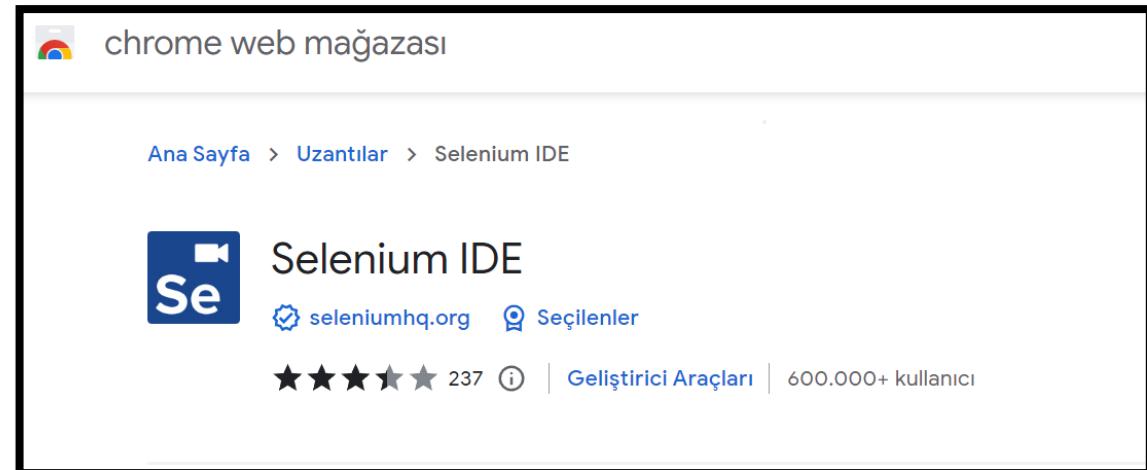
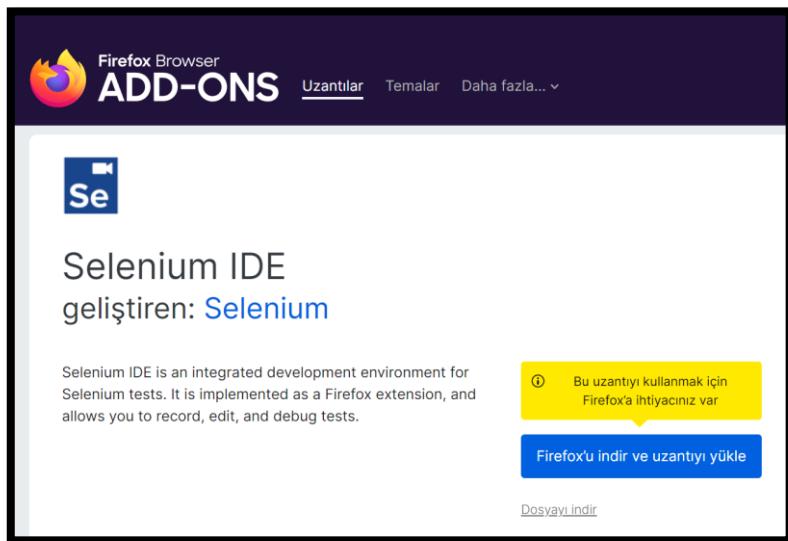


Selenium ile Test Otomasyonu Eğitimi

Ahmet GÜRBÜZ

Bu eğitime başlamadan gereken kurulumlar.

- ✓ JDK → (java -version)
- ✓ Eclipse IDE
- ✓ Selenium Driver (Kullandığınız Tarayıcıya göre değişmektedir.)
- ✓ Selenium IDE



Selenium Nedir ?



- ✓ **Selenium,farklı browserlar ve platformlarda web uygulamanızı doğrulamak için kullanılan açık kaynak kodlu bir test framework'tür.**
- ✓ **Selenium, kullanarak test scriptleri oluşturmak için bir çok programala dili kullanılır. (Java , Python, C# vb.)**

Selenium automates browsers. That's it!

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

About Selenium

Selenium is a suite of tools for automating web browsers.

Özetle

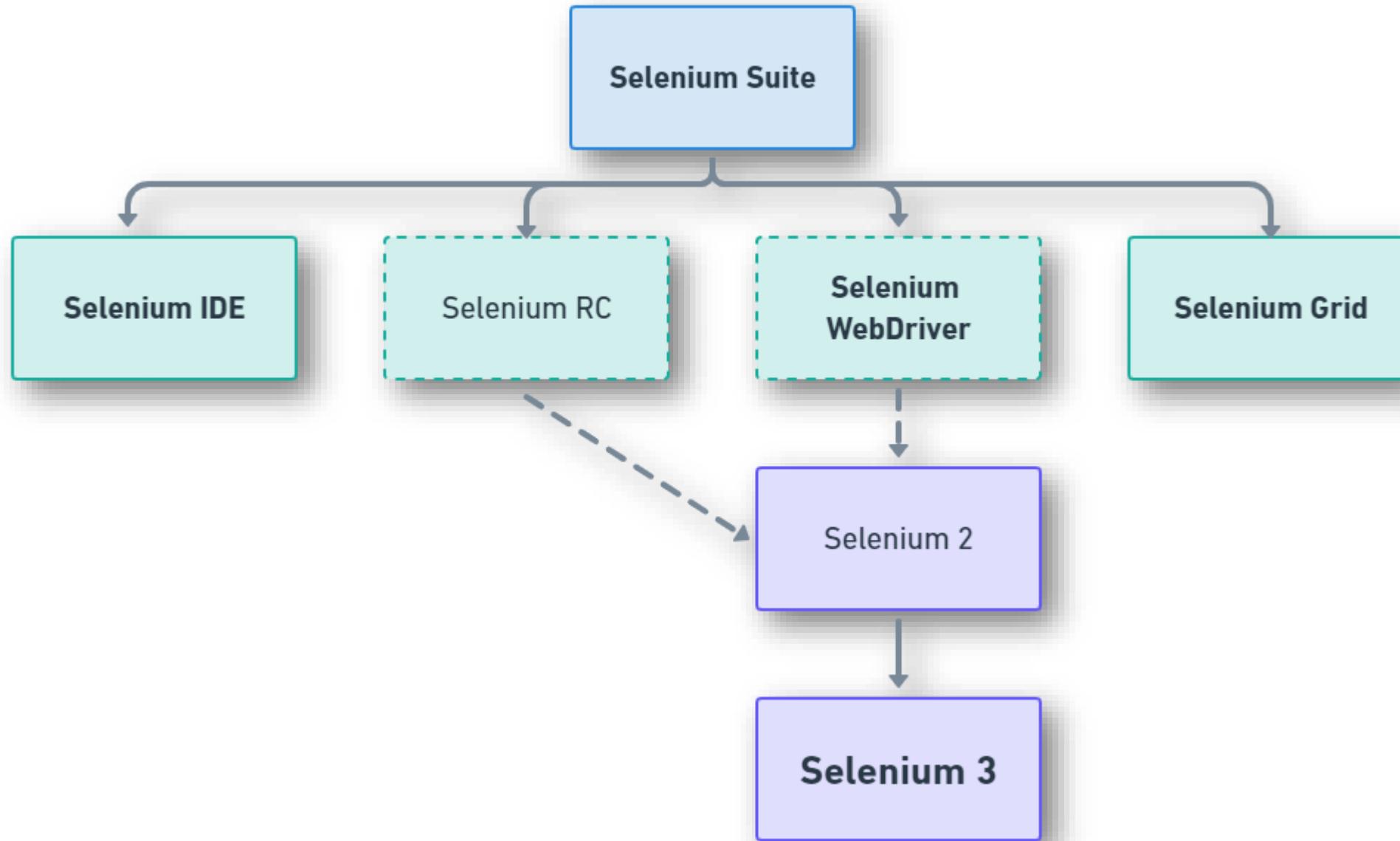
- ✓ Web Tarayıcıları için **Otomasyon Aracı**: Selenium
- ✓ Web Tarayıcıları Kontrol Eder.
- ✓ Kullanıcı Aktivitelerini Taklit Eder:
 - Yazı Yazmak
 - Drop-Down Seçimi
 - Checkbox
 - Bağlantı Tıklama
 - Fare Hareketleri
 - JavaScript Koşturma

Yani Selenium, kullanıcı neyi görüyorsa ne ile etkileşime nasıl geçebiliyorsa aynışını yapar ve taklit eder.

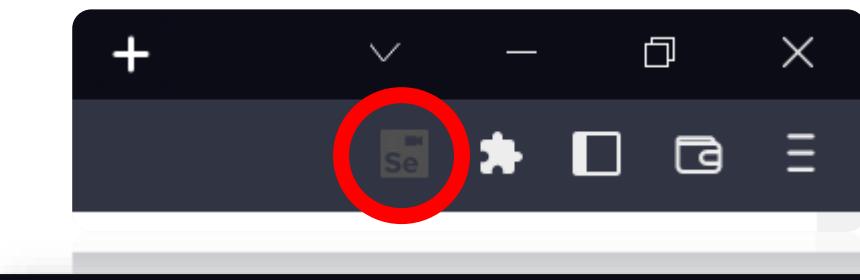
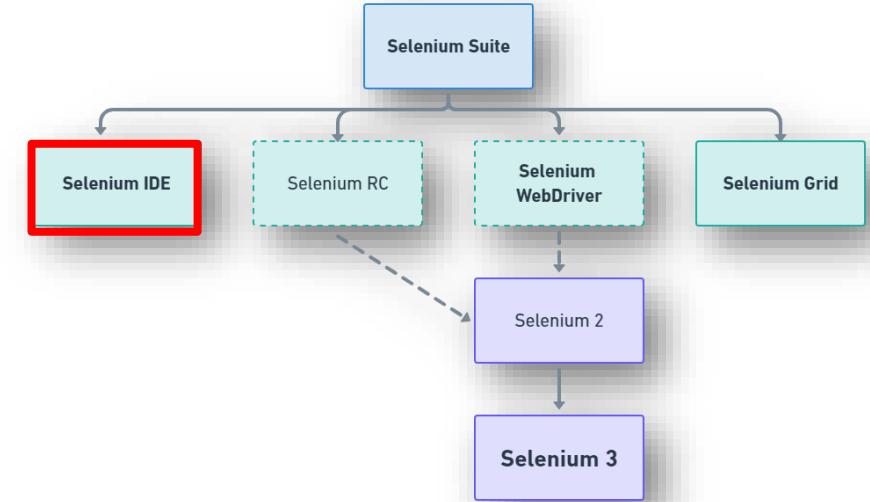
Neden Selenium kullanılıyor ?

- ✓ Bir çok dil desteği.
- ✓ Açık kaynak kodlu, Ücretsiz.
- ✓ Bir çok browser desteği.
- ✓ Geniş bir OS (Operating System) desteği.
- ✓ Öğrenimi ve kullanımı kolay.
- ✓ Başka araçlar kütüphaneler ve framework'ler ile entegre kullanımı mümkün.
- ✓ Tarayıcılarının farklarını ortadan kaldırarak aynı kodlar ile farklı tarayıcılarda aynı işi yapmanıza olanak sağlar.
(Yani tarayıcıların kendi ve özel API özelliklerini bilmeye gerek duyulmaz.)

Gelinin Noktada: Selenium' un Yapısı



Selenium IDE

A screenshot of the Selenium IDE application window showing a welcome dialog box. The dialog box has a blue header with the text "Welcome to Selenium IDE!" and "Version 3.17.2". Below the header is a question "What would you like to do?". There are four options listed: "Record a new test in a new project", "Open an existing project", "Create a new project", and "Close Selenium IDE". At the bottom of the dialog box is the text "To learn more on Selenium IDE and how to use it visit the [the Selenium IDE project page](#)". The background of the application shows the standard Selenium IDE interface with tabs for "Project", "Tests", and "Search tests...".

Selenium IDE

- ✓ İndirmek için ;
- ✓ <https://www.selenium.dev/selenium-ide/>
- ✓ Adresini kullanabilirsiniz. Chrome ve Firefox tarayıcısına doğrudan eklenti olarak indirebilirsiniz.

Selenium IDE

Open source record and playback test automation for the web



CHROME DOWNLOAD



FIREFOX DOWNLOAD



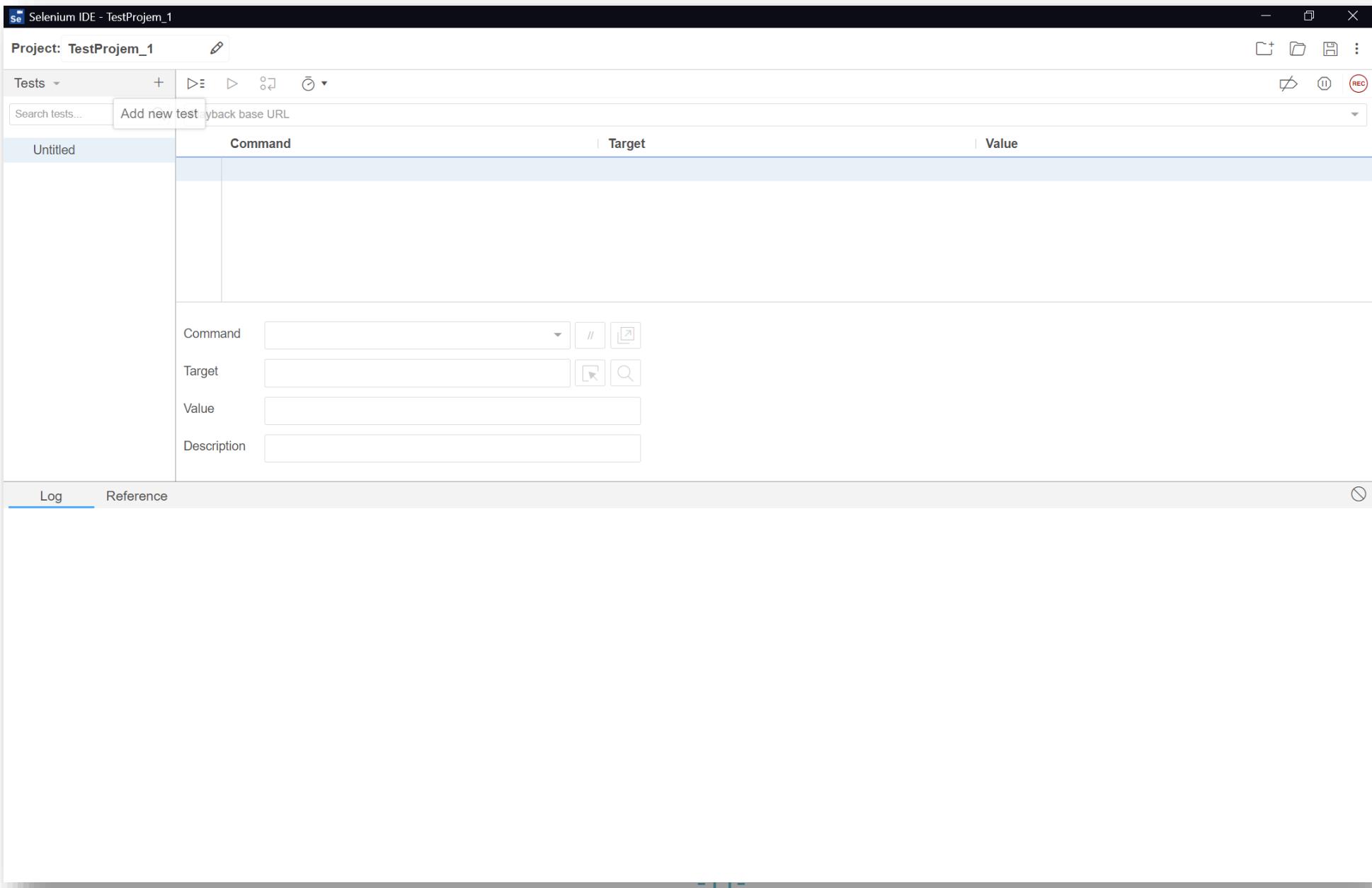
LATEST ZIP



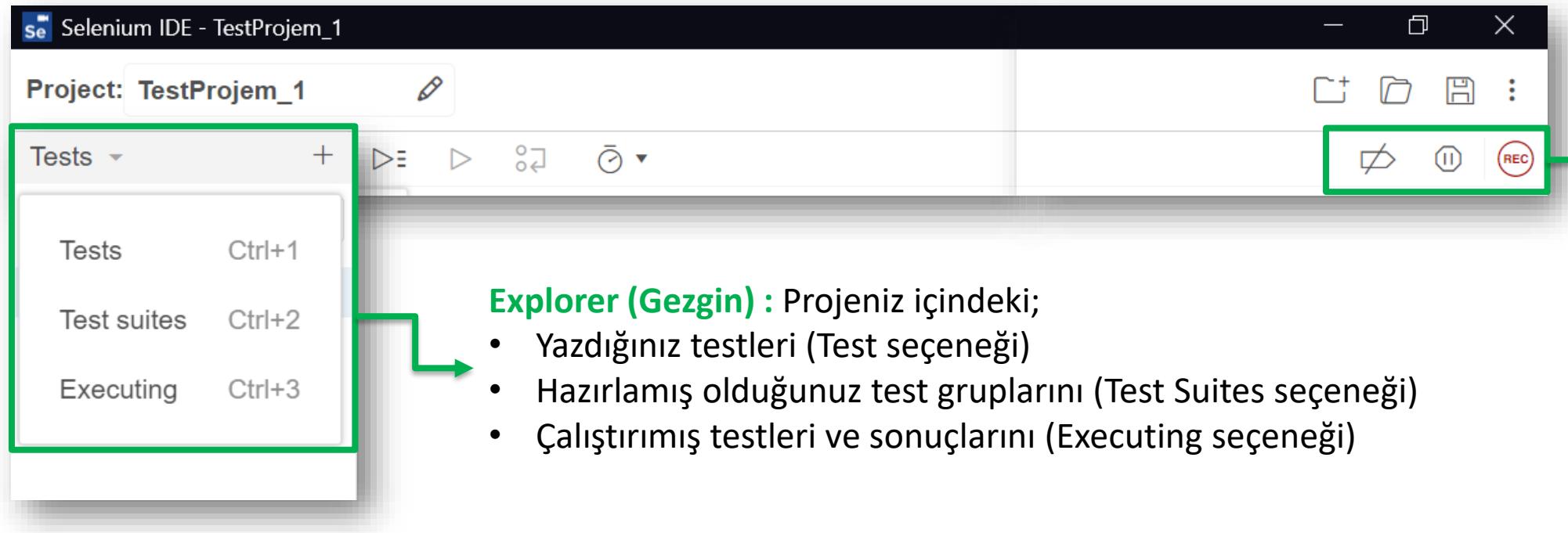
Star

2,182

Selenium IDE Panelleri



Selenium IDE Üst Paneller



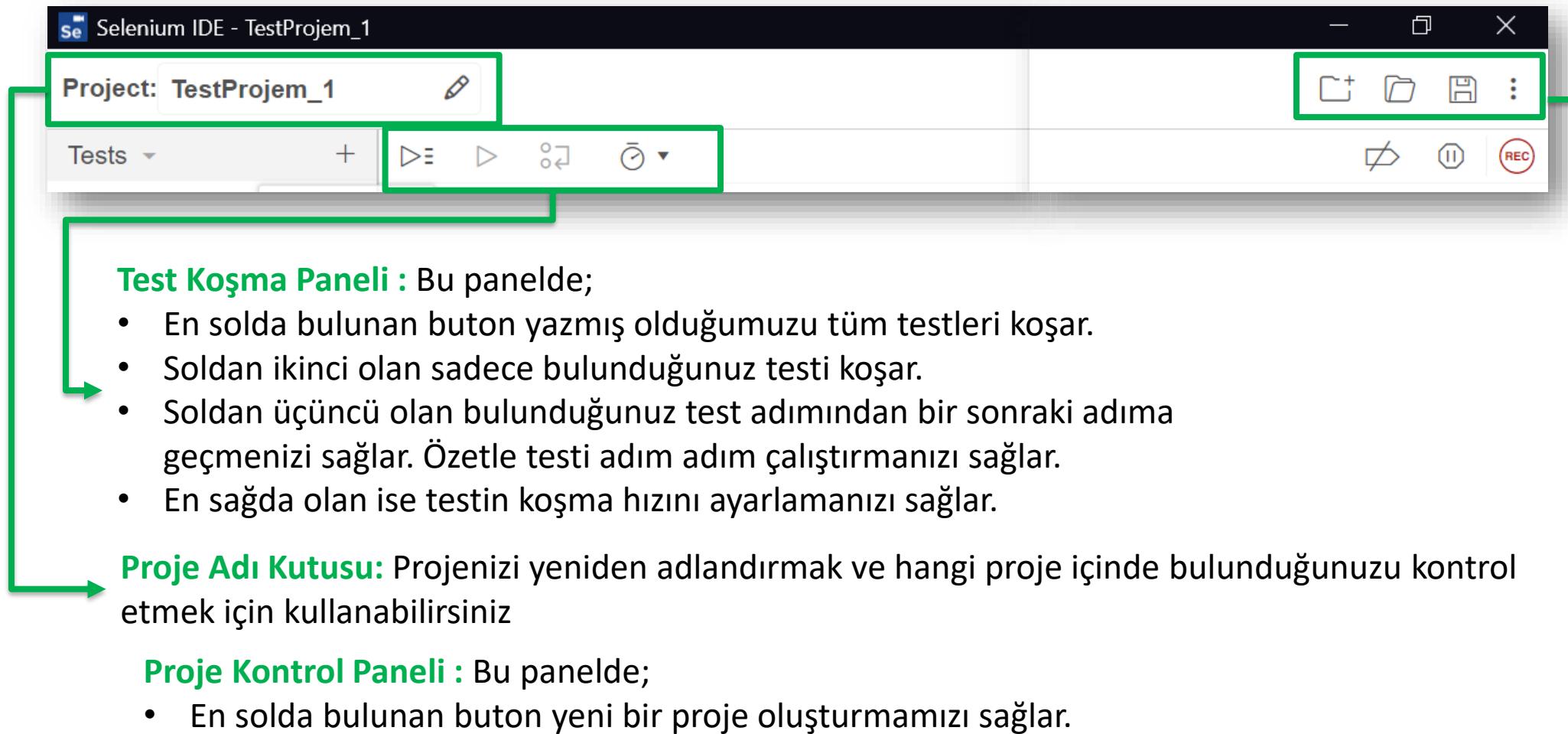
Explorer (Gezgin) : Projeniz içindeki;

- Yazdığınız testleri (Test seçeneği)
- Hazırlamış olduğunuz test gruplarını (Test Suites seçeneği)
- Çalıştırılmış testleri ve sonuçlarını (Executing seçeneği)

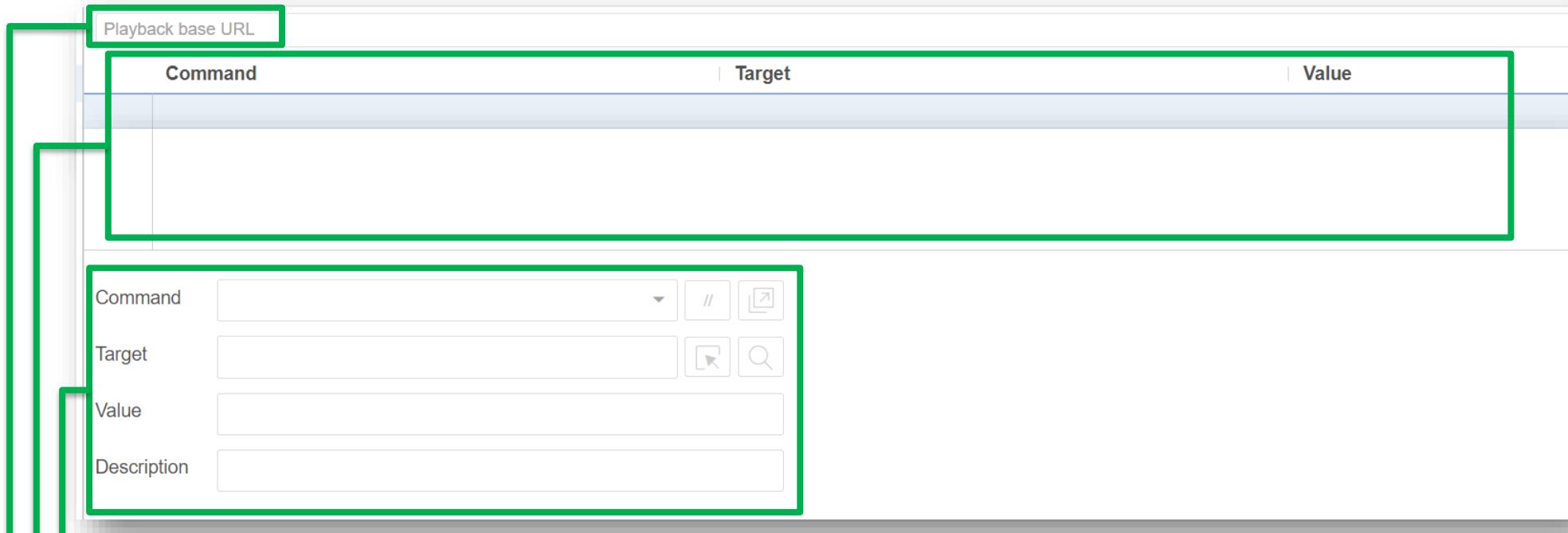
Test Kayıt Paneli :

- Soldan ilk buton eklemiş olduğumuz tüm bookmark işaretlerini etkisiz hale getirir.
- Soldan ikinci olan buton hata halinde test koşusunu sonlandırmak yerine pause mode geçmesini sağlar.
- Soldan 3. buton ise bir test kaydını başlatmanızı yada sonlandırmamanızı sağlar.

Selenium IDE Üst Paneller



Selenium IDE Orta Panel



Komut Kontrol Paneli: Test adımlarınızdaki komutlar üzerinde değişiklik yapmanızı yada yeni bir test adımı eklemenizi sağlayacak olan paneldir.

Test Adımları: Test adımlarınızın görüldüğü ana paneldir. Burada kayıt etmiş olduğunuz test adımlarını sırasıyla;

- Komut (yapılmasını istediğiniz işlem)
- Hedef (hangi dom elemanına etki etmesini istiyorsunuz)
- Değer (eğer bir değişiklik yada kontrol yapmak istiyorsanız sizin vereceğiniz değer)

Test Edilecek Ana Url: Projenizde test etmek istediğiniz ana dizinin urlidir.

Selenium IDE Alt Panel

Log Reference 

Running 'test1' 14:43:16

1. open on / **OK** 14:43:16
2. setWindowSize on 1440x900 **OK** 14:43:16
3. type on name=q with value merhaba **OK** 14:43:16
4. click on css=.LLD4me **OK** 14:43:17
5. click on css=.SDkEP **OK** 14:43:17
6. close **OK** 14:43:17

'test1' completed successfully 14:43:17

Log Reference 

type locator, value

Sets the value of an input field, as though you typed it in. Can also be used to set the value of combo boxes, check boxes, etc. In these cases, value should be the value of the option selected, not the visible text. Chrome only: If a file path is given it will be uploaded to the input (for type=file), NOTE: XPath locators are not supported.

arguments:

- locator - An element locator.
- value - The value to input.

Reference Paneli: Seçmiş olduğunuz test adımlarındaki komutların açıklamasını parametrelerini ve dökümantasyonunu gösteren paneldir. İsterseniz seleniumun kendi web sayfasına giderekde seçmiş olduğunuz komutun dokümanındaki açıklamasını bulabilirsiniz.

<https://www.selenium.dev/selenium-ide/docs/en/introduction/getting-started>

Log Paneli: Test adımlarınızda yapılan işlemleri sonuçlarını ve yapıldığı zaman damgasını gösteren paneldir.

Ayrıca uyarıların ve hataların gösterilmesinde de kullanılır. Sağ üstteki butonu kullanarak bu paneli temizleyebilirsiniz.

Selenium IDE özelliklerini tanıyalım

1. Öncelikle bir proje oluşturalım.
2. Bir test oluşturalım.
3. Bir test kaydı yapalım.
4. Testimizi koşalım ve sonuçları inceleyelim.

Selenium IDE Target Locators

1. ID
2. Name
3. Xpath
4. Css locators
5. Dom

Locator	Kod
ID	<code>id=ElementinIDsi</code> Örn: id=new
Name	<code>name=ElementinName</code> Örn: name=box
Xpath	<code>//TAG[@Attribute]</code> Örn: //div[@class='sort']
Css	Örn: css=#id
DOM	<code>dom=document.getElementById("ID")</code>

Örnek 2

Test Adımları

1. <https://demoqa.com/> adresine git.
2. Ekrandan Elements Panelini seç.
3. Açılan menüden Radio Button'ı seç.
4. Açılan ekranada no radio buttonun disabled olduğunu doğrula
5. Açılan ekranada no radio buttonun işaretli olmadığını olduğunu

Siz Deneyin

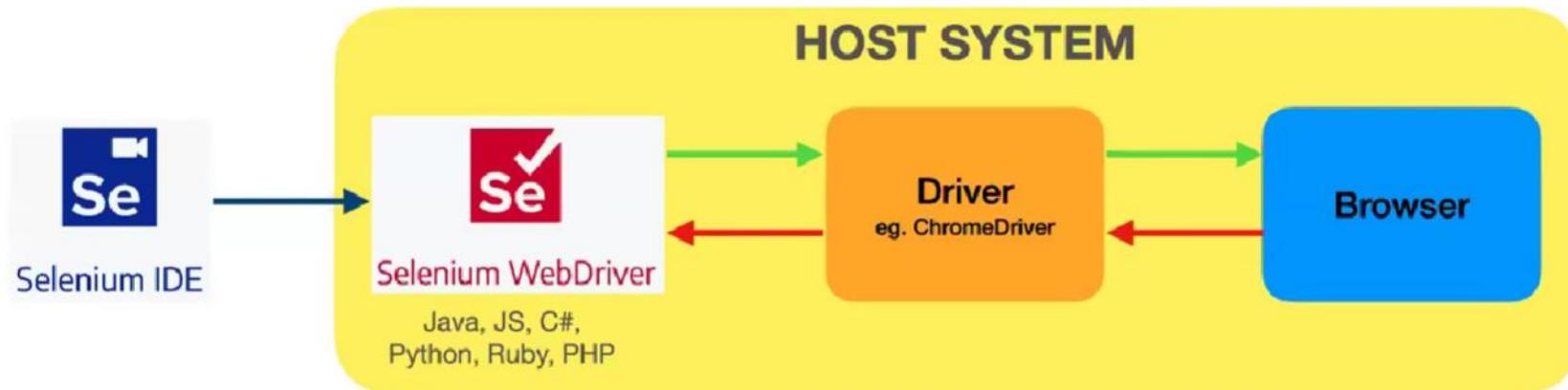
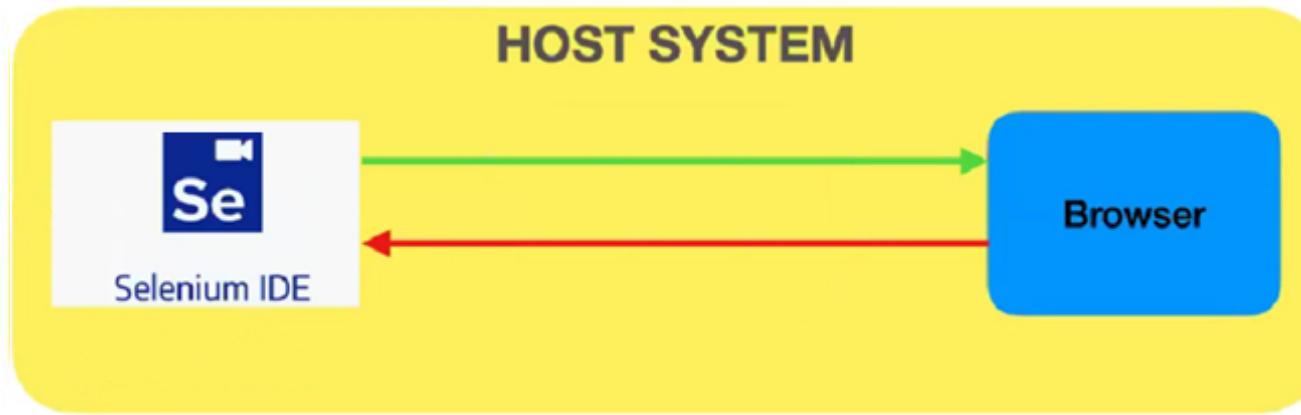
Test Adımları

1. <https://demoqa.com/> adresine git.
2. Ekrandan Elements Panelini seç.
3. Açılan menüden Check Box seç.
4. Açılan ekranda Desktop ve Documents checkbox'ını onayla.
5. You have selected bölümünde bu seçilen bölümlerin olduğunu doğrula.

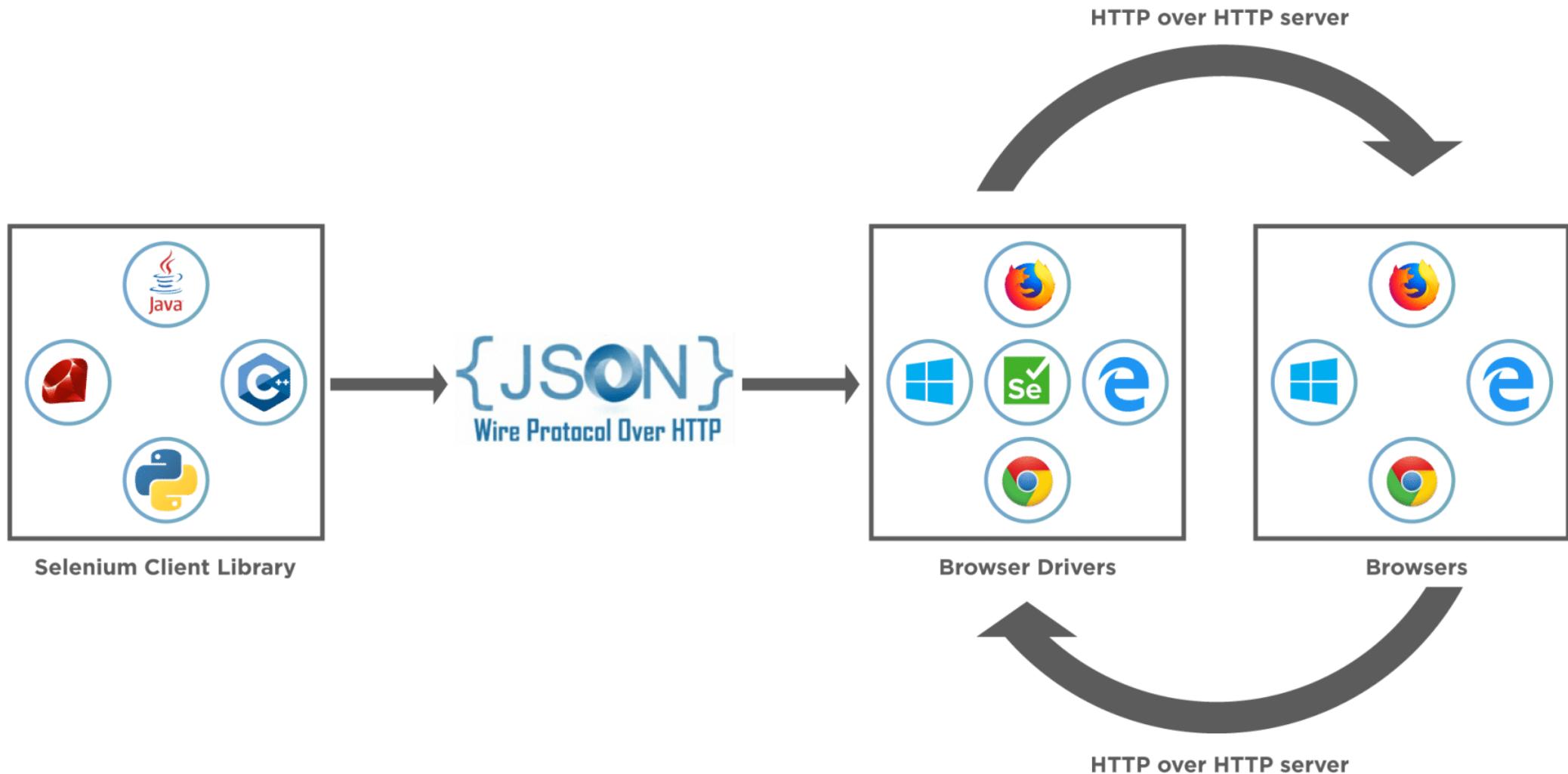
Peki neden Ide – Driver var, Farkları neler ?

Selenium WebDriver	Selenium IDE
Bir API Kütüphanesidir.	Bir tarayıcı eklentisidir. Test adımlarını kaydeder ve tekrar oynatır.
Daha fazla bağımlılık içeren daha fazla ayar gerektiren bir kurulumu vardır.	Tek tıkla eklenti olarak kurulabilir.
En az bir dilde kodlama bilgisine ihtiyaç duyulur.	Kodlama becerisi gerektirmez.
Kodlama becerinizi ve başka Tool – Kütüphaneler - Framework beraber kullanabilirsiniz.	Doğrudan başka Tool – Kütüphaneler ile entegre edemezsiniz.
Herhangi bir tarayıcı ile birlikte çalışabilir. Tarayıcıya bir nesne gibi davranır.	Firefox ve Chrome ile birlikte kullanılabilir.

Selenium IDE-WebDriver Yapısı ?



Selenium WebDriver Nedir ?



Selenium WebDriver’ı kullanmak için...

✓ Programlama dili

- Selenium ile çalışırken kodlarımızı yazabileceğimiz bir programlama dili kullanmamız gereklidir. Biz **Java** kullanacağımız.

✓ IDE

- Kodlarımızı çalıştırılmak için bir compiler'a ihtiyacımız olacak. Biz **IntelliJ** kullanacağımız.

✓ Derleme Aracı (Build Tool)

- Biz aslında bir proje oluşturduğumuz için bu proje içinde kullandığımız tool'ların yüklenmesi, güncel tutulması ve istenen yardımcı eklentilerin projemize dahil edilebilmesi için bir derleme aracına ihtiyaç duyuyoruz. Biz Maven kullanacağımız.

✓ Özetlersek

- **IntelliJ'de** bir **Maven** projesi oluşturup, Maven'in bize sağladığı dependency(kütüphaneler) ile projemize **Selenium WebDriver** ve ihtiyacımız olacak yardımcı kütüphaneleri ekleyip, **Java** ile bu kütüphanelerdeki HAZIR METHOD'ları kullanacağımız.

Maven Nedir ?

- ✓ **Maven, neden gereklidir?** Java projesini oluşturmak ve yönetmek için kullanabileceğimiz bir araçtır.
 - Bu araç, projemizin kütüphanelerini GitHub repository'lerini sürekli kontrol ederek, yapılan update'leri projemize taşır.
 - Bunun için GitHub repository'leri ile projemiz arasında bağlantı kuran basit Maven'ın amacını biliyor musunuz? (dependency)
 - ✓ **Maven'in amacı nedir?** Maven, aşağıdaki konulara odaklanmıştır :
 - Maven yapısı ile genel bir işleyişinin nasıl geliştiğimizi göstermek, projelerin kütüphanelerinin depolandığı ve güncellendiği yerde kayıtlı hale getirmek. Bu etkileşefektlaşmak için Maven, aşağıdaki konulara odaklanmıştır :
- NOT:** Projelerin versiyonları güncel olmalıdır, stabil versiyon ve kullanım sayısı dikkate alınmalıdır. Projenin güncelliliğinin kolayca sağlanması.
- Tüm developer'lar için ortak bir build sistemi sağlamak.
 - Proje hakkında doğru ve anlaşılabılır bilgilere hızlıca erişimin sağlanması.

Hadi bir proje oluşturalım...

Pom Dosyası ve Proje Oluşturmak

- ✓ Project Object Model dosyası.
- ✓ İhtiyacımız olan paketleri buradan ekleyebiliriz.
- ✓ Öncelikle ekleme istediğimiz paketleri <dependencies> tagi arasına paketlerimizi ekleyebiliriz.
- ✓ Paketleri bulmak için <https://mvnrepository.com/> adresine gidip paket araması yapabiliriz.

Web Driver – Proje

- ✓ İster manuel ister otomatik olarak webDriver’ı projemiz içine ekleyebiliriz.
- ✓ İkisinin de avantajları ve dezavantajları bulunmaktadır.
- ✓ 3 Farklı yol ile projemize webDriver’ı eklemek mümkün.
- ✓ **Birincisi** Eğer ki Windows kullanıcısı iseniz.
System32 klasörü içine driveri atabilirsiniz.
- ✓ **İkincisi** Manuel olarak her projeye driveri elle ekleyebilirsiniz.
- ✓ **Üçüncüüsü** [WebDriverManager](#) ile sistemimize doysa indirmeden işletim sisteminde bağımsız olarak projemizi çalıştırabiliriz.

Web Driver – Proje

Quick Reference

Browser	Supported OS	Maintained by	Download	Issue Tracker
Chromium/Chrome	Windows/macOS/Linux	Google	Downloads	Issues
Firefox	Windows/macOS/Linux	Mozilla	Downloads	Issues
Edge	Windows/macOS	Microsoft	Downloads	Issues
Internet Explorer	Windows	Selenium Project	Downloads	Issues
Safari	macOS High Sierra and newer	Apple	Built in	Issues

Başlamaya Hazırız, Ufak Birkaç Hatırlatma !

- ✓ Eğer ki system32 içine atmadiysak kodlamaya başlarken exe dosyasının lokasyonunu ayarlamamız gerekiyor.

```
System.setProperty("webdriver.chrome.driver", "Dosya konumu");
```

- ✓ Dip Not: Eğer ki Chrome değil. Ama Chrome tabanlı başka bir browser kullanıyorsanız. Örn: Brave

```
ChromeOptions options = new ChromeOptions().setBinary("Dosya konumu");
WebDriver driver = new ChromeDriver(options);
```

Öncelikle Driver’ı tanıyalım

- ✓ Driver aslında bizim web tarayıcısını temsil ettiğimiz objemizdir.
- ✓ Nasıl bir kullanıcı tarayıcı üzerinde etkileşim butonları kullanabiliyor, (sayfayı büyüt, geri git, ileri git, sayfayı yenile, sayfayı getir vb..) tarayıcının özelliklerini değiştirebiliyorsa bizde bu obje sayesinde aynı işlemleri yapıyor olacağız.
- ✓ Bu sebeple bu obje üzerindeki en çok kullanılan önemli fonksiyonları bilmek önemlidir.

Selenium ile uygulamalarda navigasyon

- ✓ Öncelikle navigasyon için temel metodlar.

```
//Sizin için bir sayfayı açar  
driver.get("https://selenium.dev");
```

```
// aynı işlemi yapar ama forward ve back yapılmasına imkan tanır.  
driver.navigate().to("https://selenium.dev");
```

```
//bir önceki sayfaya dönüş yapar.  
driver.navigate().back();
```

```
//back() ile geldiği sayfaya yeniden gider  
driver.navigate().forward();
```

```
//içinde olduğu sayfayı yeniler  
driver.navigate().refresh();
```

Seleniumda Önemli Driver metodları

//bulunduğumuz sayfayı maximize eder.
driver.manage().window().maximize();

// içinde olduğu sayfanın başlığını döndürür.
driver.getTitle()

//Bulunduğu sayfanın url'ini döndürür.
driver.getCurrentUrl()

// Bulunduğu sayfanın ÖZGÜN hash değerini döndürür.
driver.getWindowHandle()

//driver çalışırken açılan tüm sayfaların ÖZGÜN hash değerini döndürür.
driver.getWindowHandles()

Siz deneyin

1. 'selenium.dev' sayfasına gidin
2. Sayfanın başlığını konsolunuza yazdırın.
3. 'Google.com' sayfasına gidin
4. Sayfanın url'ini konsolunuza yazdırın
5. Bir önceki sayfaya geri dönün
6. Sayfanın başlığını konsolunuza yazdırın
7. Sayfayı yenileyin
8. Uygulamayı bitirin.

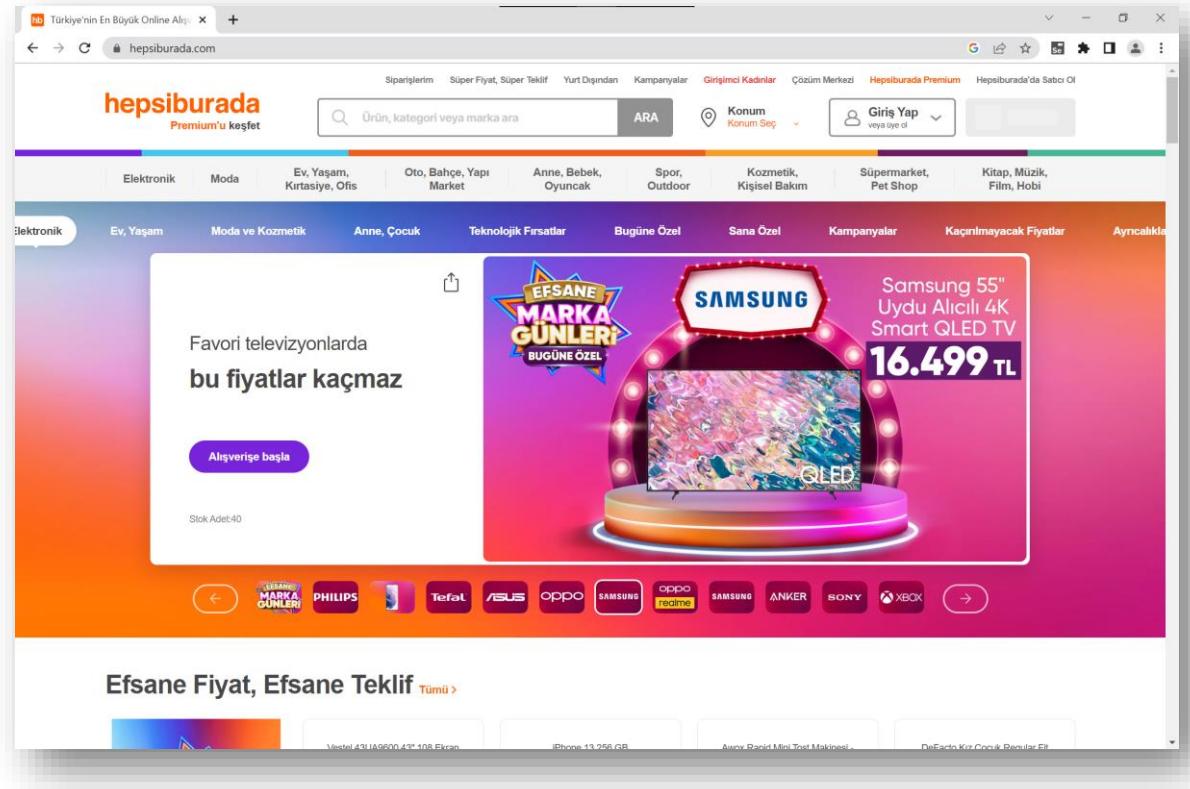
**Peki Sayfada bulunan elementler ile
nasıl etkileşime geçeceğiz ?**

HTML Neydi ?

- ✓ Hypertext Markup Language bilinir.
- ✓ Web sayfaları oluşturmak için dünya standarttır.
- ✓ Bir web sayfasının yapısını tanımlar.
- ✓ Bir dizi elemandan oluşur.
- ✓ Tarayıcıya sayfa içeriğini nasıl göstereceğini belirtir.
- ✓ Elemanları etiketler (tag) ile gösterilir.
- ✓ Tarayıcılar bu etiketleri göstermez, sayfayı içeriğini oluşturmak için kullanır.

Web Elementi

- ✓ **webElement:** Bir web uygulamasında kullanılan etkileşimli veya etkileşimsiz her şeydir.
- ✓ Selenium ile test otomasyonu yapabilmek için kullanacağımız **webElementlerini** nerede bulacağımızı Webdriver'a söylememiz gereklidir.



DOM

Document Object Model

document

Root element:

<html>

Element:

<head>

Element:

<title>

Text:

"My title"

Element:

<body>

Element:

<h1>

Text:

"A heading"

Element:

<a>

Attribute:

href

Text:

"Link text"

<html>

<head>

<title>Page title</title>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

</body>

</html>

Locators

- ✓ **Locator:** Genel olarak bir web sitesindeki herhangi web elementini WebDriver'a tarif etme yöntemidir.
- ✓ Bu tarif etme işi için sayfanın html yapısı ve bu yapının içinde bulunan Web Element'leri kullanılır.

hepsiburada

Premium'u keşfet



- ✓ Bir WebElement'inin tarif edilmesine yarayan 3 temel unsuru vardır.

```
<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 WebElement'inin Unsurları

```
<inp <script>
  name GwInstrumentation.markH1Af({ uri: "/ah/ajax/counter?_="
  clas ctr=desktop_ajax_atf&exp=13&rId=T4M&mkId=&h=4cc3to"
  tabi })
    </script>
```

- ✓ **Tag:** Bir WebElement'inin başlangıcını ve bitişini göstermek için kullanılan işaretleme kelimesidir.
- ✓ **Attribute:** Bir WebElement'i'nin davranışını kontrol etmek için tag'in yanında yazılan kelimelerdir. Bu kelimeler elementin davranışlarını değiştirebilir.
- ✓ **Attribute value:** Attribute'ların değerlerini belirlemek için kullanılır, kullanım yerine ögenin davranışını etkileyebilir.

Özetle Locator...

- ✓ Birden fazla webElement için aynı tag, attribute ve attribute value kullanılabilir.
- ✓ Locate(atma) işlemi ise Unique (benzersiz) olmalıdır.
- ✓ Eğer olmazsa driver kendisine locate edilen(atanan) elementlerden hangisine gideceğini bulamaz. Böyle bir durumda NoSuchElementException hatası ile karşılaşılır.
- ✓ Özetle en önemli nokta **Locate(atama) işlemi** benzer özellik gösterebilen **webElementlerini unique(özgün)** olarak **belirleme** işlemidir.

Selenium Locatorları / Locate Metodları

1. By CSS ID: `findElement(By.id("ID"))`
2. By name attribute : `findElement(By.name("Name"))`
3. By CSS class name: `findElement(By.className("ClassName"))`
4. By tagName: `findElement(By.tagName("tagName"))`
5.

```
<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">
```
6. By css selector: `findElement(By.cssSelector("cssSelector"))`
7. By DOM structure or Xpath: `findElement(By.xpath("Xpath"))`

Css Seçicileri

- ✓ Xpath'e benzer ve tüm web Elementler için kullanılabilir. Farkı xpath'de kullandığımız **//** ve **@** işaretinin kullanılmamasıdır.

tagName[attributelsmi='attributeValue']

Örn: div[fill='new']

- ✓ Eğer kullanılacak attribute id veya class ise pratik olarak yazılabilir
 - **#iddeğeri**
 - **.classdeğeri**

Temel Css Seçiciler

```
<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">
```

Selector	Açıklama	Kod
Tag	Html etiketini seçmek için kullanılır. Ör : div, input, span	Tag Adı
Class	Tagleri sınıf adlarını kullanarak seçmek için kullanılır.	.Sınıf Adı
ID	Tagleri id özelliğini kullanarak seçmek için kullanılır.	#id adı
Attribute	Herhangi bir özelliğini kullanarak seçim yapmak için kullanılır.	[Özellikadı='ÖzellikDeğeri']
Group	Grupça seçim yapmak için kullanılır. Seçilicilerin arasına ,(virgül) koyarak yazarız	Seçici1, seçici2, seçici3

Xpath Seçicileri

- ✓ Xpath en güçlü locator'dır.
- ✓ XPath XML Path Language anlamına gelir.
- ✓ 2 Çeşit Xpath vardır. Bunlar Relative, Absolute Xpath'dır.
- ✓ **Relative Xpath:** Kökten değil seçilmiş bir elemandan başlar. Seçilmiş bir elemanla başladığı için her zaman güvenilir bir yoldur. Tercihen istenen öğeye en yakın eleman seçilir. Seçilen elemanın seçicisinin önüne // işaretini konur.

//tagName[@attributeName='attributeValue']

Örn: //div[@class='new']/span

Xpath Seçicileri

- ✓ **Absolute Xpath:** Domun en tepesinden başlayarak bir öğeyi bulmanın yoludur.

Örn: /html/body/div/span

Xpath Seçicileri

Selector	Açıklama	Kod
Attribute	Herhangi bir özelliğini kullanarak seçim yapmak için kullanılır.	//TAG[@Attribute] Örn: //div[@class='new']/span
Child	Bir tag'in child elemanlarını seçmek için kullanılır.	//AnaSeçici/child::Seçici2 Örn: //div[@class='new']/child::span
Parent	Seçilen elementin parent elementine ulaşmak için kullanılır.	//TAG[@Attribute]/.. Örn: //div[@class='sort']/..
Text()	Seçilen element içindeki texti almak için kullanılır.	//TAG[text()='TEXT1']

Daha Fazla Detay İçin Bakınız : <https://www.browserstack.com/guide/xpath-locators-cheat-sheet>

Hadi Bunları birleştirelim...

- ✓ Öncelikle En başa dönemim ve tekrar soralım test otomasyonu nedir ?
 - Test otomasyonu manuel olarak gerçekleştirebildiğimiz herhangi bir test senaryosunu Selenium driver kullanarak kodlarla gerçekleştirmektir.

Hadi Bunları birleştirelim...

- ✓ Test Otomasyonu nasıl yapılır ?
 1. Öncelikle web elementini görsel olarak belirleyeceğiz.
 2. Elementi inspect ederek elementin html kodunu görüntüleyeceğiz.
 3. Elementi unique olarak seçebileceğimiz bir locator yazacağız.
 4. Selenium içine daha önce görmüş olduğumuz locate metodalarımız yardımı ile bu seçimi aktaracağız.
 5. Daha sonra web elementi üzerindeki metodları kullanarak gerekli aksiyonu gerçekleştiriyoruz.
 6. Doğrulama verisini ekranдан alıp doğrulamayı yapıyoruz.

Beraber Yapalım - Test senaryosu – 0

1. <https://www.ebay.com/> adresine git.
2. Açılan ekranında arama çubuğuuna Airpods yaz.
3. Enter'a basarak (submit ederek) aramayı yap.
4. Airpods araması yapıldığını doğrula.

Test senaryosu - 1

Test Adımları

1. <https://demoqa.com/login> adresine git.
2. Açılan ekranda UserName ve Password bölümlerini doldur.
3. Login butonuna bas.
4. Giriş yapılamadığını doğrula.

Test Verileri

UserName : newdemo

Password: Newdemotester@123

Test senaryosu - 2

Test Adımları

1. <https://demoqa.com/login> adresine git.
2. Açılan ekranınUserName ve Password alanlarını doldur.
3. Login butonuna bas.
4. Giriş yapıldığını doğrula

Test Verileri

UserName : newdemotester

Password: Newdemotester@123

Test senaryosu - 3

Test Adımları

1. <https://demoqa.com/text-box> adresine git.
2. Açılan ekranın Full Name, Email, Current Address, Permanent Address alanlarını doldur.
3. Submit butonuna bas.
4. Ekrana gelen text bölümündeki bilgilerin test verileri ile aynı olduğunu doğrula.

Test Verileri

Full Name: Ahmet

Email: ahmet@ben.com

Current Address : İstanbul

Permananet Address : İzmir

Selenium Waits Kavramı

- ✓ Genel olarak neden beklemeye ihtiyaç duyuyoruz ?
- ✓ Bazen otomasyonumuz tarayıcıımızın yükleme hızından daha hızlı çalışabilir. Böyle bir durumda no such element hatası alınır.
- ✓ Farklı elementler farklı yükleme zamanlarında yüklenebilir.
- ✓ Webdriver scriptleri ile browserin eventleri her zaman senkron olamayabilir.

Implicit Wait (Global Wait)

- ✓ Kullandığımız sitelerde genel olarak asenkron ve başka sitelerden alınan yapılarımız olabilir. Yada bağlantımız yavaş olabilir.
- ✓ Böyle durumlarda ulaşmak istediğimiz elementi beklememiz gerekebilir.
- ✓ implicitlyWait driver'a sayfanın yüklenmesi ve kullanacağımız web elementlerin bulunması için bekleyeceği maximum süreyi belirtir.
- ✓ driver bu süre içerisinde sayfa yüklenir/web element bulunursa beklemeden çalışmaya devam eder.
- ✓ bu süre bittiği halde sayfa yüklenmemiş/webElement bulunamamışsa exception vererek çalışmayı durdurur.
- ✓ Driver tüm UI elementlerini beklediği için global wait olarak bilinir.

Implicit Wait (Global Wait)

- ✓ Bu bekleme driver üzerine atanır ve driver kapatılana kadar bu ayar kullanılmaya devam eder. Verilen süre içerisinde her 500ms de bir selenium sayfanın durumunu kontrol eder.
- ✓ Özette implicitly wait sayfadaki tüm elementlerin yüklenmesini verilen sürede kibarca bekle demek diyebiliriz.
- ✓ Default olarak bu değer normal şartlarda 0 dır.

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

Explicit Wait (Dynamic Wait)

- ✓ Bazı koşullar karşılanması için beklememiz gerekiğinde kullanılmalıdır.
- ✓ Bizim seçmiş olduğumuz tek bir web elementi için gerçekleştirilir.
- ✓ Bekleme süresini bu sebeple WebElemanına göre ayarlayabiliriz.
- ✓ Bu tür beklemelere bu sebeple Dynamic Wait denir.

- ✓ Driverdan belirli bir webElemanı için belirli bir koşulu belirli bir süre beklemesini isteriz. Bu süre içerisinde koşul gerçekleşirse berinlen süre tamamen dolmadan işlem devam eder. Eğer ki bu sürede gerçekleşmez ise hata döndürür.

Explicit Wait (Dynamic Wait)

1. WebDriverWait wait = new WebDriverWait(driver, 30);
 2. wait.until(ExpectedConditions.elementToBeClickable(selected_button))
 3. selected_button.click();
-
1. WebDriverWait wait = new WebDriverWait(driver, 30);
 2. WebElement myButton =
wait.until(ExpectedConditions.elementToBeClickable(By.id("id")));
 3. myButton.click();

NOT: Explicit Wait içinde de implicit wait gibi verilen süre içerisinde her 500ms de bir selenium koşulun durumunu kontrol eder.

Uyarı...

- ✓ Eğer Explicit ve implicit wait'i karışık olarak kullandıysak bir teste toplam bekleme zamanını düşünmemiz iyi olacaktır. Şöylediki;

Implicit wait : 10 saniye

Explicit wait : 15 saniye

Toplam bekleme zamanı max 25 saniye ye kadar çıkabilir. Böyle bir durumda aslında siz o koşula toplamda 15 saniye vermek isterseniz implicit wait süresinin üstüne bu süreyi ekliyebilirsiniz. → Explicit wait : 5 saniye

Örnek yapalım

1. https://the-internet.herokuapp.com/dynamic_controls adresine gidin.
2. Remove butonuna basın.
3. “It’s gone!” mesajının görüntünlendiğini doğrulayın.
4. Add butonuna basın
5. “It’s back” mesajının göründüğünü test edin

Örnek yapalım;

1. <https://demoqa.com/dynamic-properties> adresine gidin.
2. Will enable 5 seconds'in etkin olmadığını(enabled) test edin.
3. Will enable 5 seconds etkin oluncaya kadar bekleyin ve enabled olduğunu test edin.
4. Açılan ekranın Will enable after 5 seconds butonunu basın.
5. Butonun etrafındaki Hover'ın yandığını görün.

Fluent Wait

- ✓ Explicit wait benzer. Bazı koşullar karşılanması dek beklememiz gerekiğinde kullanılabilir.
- ✓ Farkı verilen süre içerisindeki kontrol aralığını özgürce değiştirebiliriz. (Polling Frequency)
- ✓ Ayrıca koşulun karşılanması olmadığı caslerdeki hatayı ignore ettirebiliriz.
- ✓ Yani **Explicit Wait ve implicit wait içinde olduğu gibi** verilen süre içerisinde her sabit olarak 500ms de bir selenium koşulun durumunu **kontrol etmesi gerekmez**.

Fluent Wait

```
Wait<WebDriver> fluentWait = new FluentWait<WebDriver>(driver)
    .withTimeout(Duration.ofSeconds(10))
    .pollingEvery(Duration.ofSeconds(1))
    .ignoring(NoSuchElementException.class);

WebElement id = fluentWait.until(ExpectedConditions.elementToBeClickable(By.id("id")));
id.click();
```

- ✓ NOT: NoSuchElementException sınıfı org.openqa.selenium paketi içinden import edilmeli bazı durumlarda import hatası görülebiliyor.

Junit (Framework) Nedir ?

JUnit, Java programlama dili için **unit test framework**'u olarak bilinir.

Unit Test: Unit Test, bir yazılımın en küçük test edilebilir parçasının, tek başına ve bağımsız olarak, test edilmesidir.

Framework, üzerine yazılım oluşturabileceğiniz bir altyapıdır. Framework, oluşturacağımız proje için bir temel görevi görür, böylece sıfırdan başlamaz, nasıl bir yapı oluşturmalıyım diye düşünmezsiniz.

JUnit Framework'unun avantajları nelerdir ?

1. Açık kaynaklıdır.
2. Test method'larini bağımsız olarak tanımlamak ve çalıştırırmak için anotasyonlar(**annotations**) sağlar.
(Böylece main method'a ihtiyacımız kalmaz)
3. Expected Result ve Actual Result'i karşılaştırarak Test etmemizi ve testin sonucunu otomatik olarak almamızı sağlayan hazır assertion method'ları barındırır ve anında geri bildirim sağlar.
(Böylece if-else ile test yapma, sonuçları yazdırma gibi işlemlere ihtiyacımız kalmaz)
4. JUnit, test düzgün çalışıyorsa **yeşil** olan bir çubukta test ilerlemesini gösterir ve bir test başarısız olduğunda **kırmızıya** döner.

Junit Annotations

- ✓ Notasyonlar variable, method, class veya package için kullanılabilir.
- ✓ Notasyonlar kullanıldığı bileşene ilave özellikler kazandırır.
- ✓ Notasyonlar ile derleyiciye(compiler) talimatlar verebiliriz.
- ✓ JUnit ile en çok kullanılan notasyonlar;

@Test

@BeforeAll @AfterAll

@BeforeEach , @AfterEach

@Disabled

@Order, @DisplayName

Assertion Class’ı

- ✓ JUnit assertion(verify/doğrulama) için Assertion class’ından static method’lar kullanırız.
- ✓ Bu method’ları kullandığımızda sonuçları if-else ile kontrol etmemize ve durumu raporlamak için yazdırma işlemine gerek kalmaz.
- ✓ Junit, test senaryomuza uygun assertion(verify/doğrulama) yapabilmemiz için farklı method’lar barındırır.

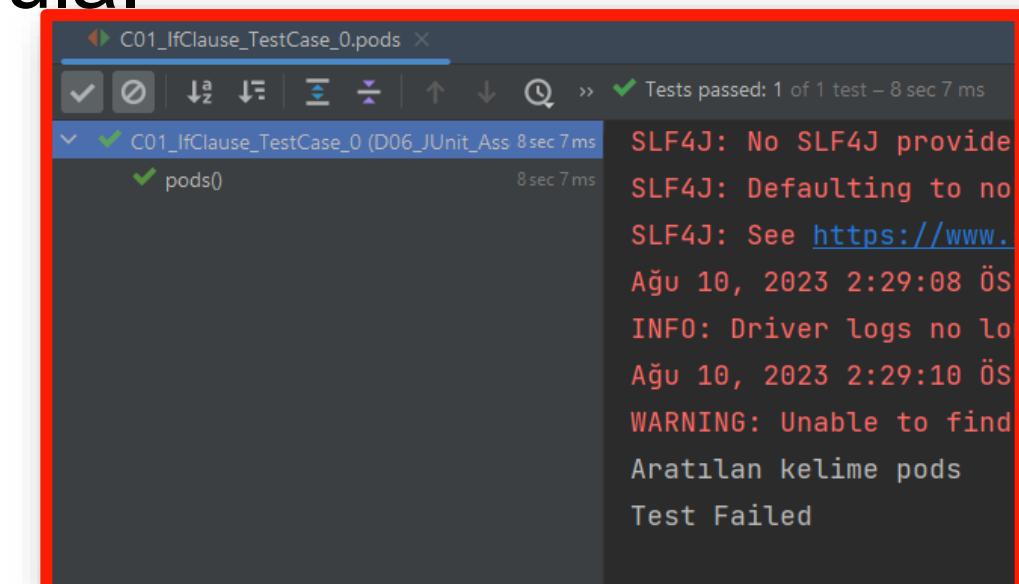
If clause Assertion Bu testi çalışıralım

1. <https://www.ebay.com/> adresine git.
2. Açılan ekranında arama çubuğuuna Airpods yaz.
3. Enter'a basarak (submit ederek) aramayı yap.
4. Airpods araması yapıldığını doğrula.

If clause Assertion

✓ Aşağıdaki testi çalıştırduğımızda nasıl bir sonuç bekleriz.

1. <https://www.ebay.com/> adresine git.
2. Açılan ekranda arama çubuğuna pods yaz.
3. Enter'a basarak (submit ederek) aramayı yap.
4. Airpods araması yapıldığını doğrula.



The screenshot shows a Java IDE interface with a dark theme. At the top, there's a toolbar with various icons. Below it is a status bar displaying "Tests passed: 1 of 1 test – 8 sec 7 ms". The main area shows a tree view of test cases under the project "C01_IfClause_TestCase_0.pods". One test case, "C01_IfClause_TestCase_0 (D06_JUnit_Ass)", is expanded, showing a single method "pods()". To the right of the tree view, there's a log window with the following output:

```
SLF4J: No SLF4J provider  
SLF4J: Defaulting to no  
SLF4J: See https://www.  
Ağu 10, 2023 2:29:08 ÖS  
INFO: Driver logs no lo  
Ağu 10, 2023 2:29:10 ÖS  
WARNING: Unable to find  
Aratılan kelime pods  
Test Failed
```

JUnit Framework'unun avantajları nelerdir ?

1. Açık kaynaklıdır.
2. Test method'larini bağımsız olarak tanımlamak ve çalıştırırmak için anotasyonlar(**annotations**) sağlar.
(Böylece main method'a ihtiyacımız kalmaz)
3. Expected Result ve Actual Result'i karşılaştırarak Test etmemizi ve testin sonucunu otomatik olarak almamızı sağlayan hazır assertion method'ları barındırır ve anında geri bildirim sağlar.
(Böylece if-else ile test yapma, sonuçları yazdırma gibi işlemlere ihtiyacımız kalmaz)
4. JUnit, test düzgün çalışıyorsa **yeşil** olan bir çubukta test ilerlemesini gösterir ve bir test başarısız olduğunda **kırmızıya** döner.

Assertion Class'ı

- ✓ En çok kullanılan assertion method'ları :
 - 1) Assertions.assertTrue(**koşul**)
 - 2) Assertions.assertFalse(**koşul**)
 - 3) Assertions.assertEquals(**expected, actual**)
 - 4) Assertions.assertAll(...)
- ✓ **NOT:** Hangi method'u seçeceğimize bize verilen değerlere göre değil,
Test senaryosunun neyi öngördüğüne göre karar veririz.

Asserts

- ✓ Örnek 1 : dönen değerin “Ahmet” içerdigini test edin

Assertions.assertTrue(donendeger.contains("Ahmet"))

- ✓ Örnek 2 : dönen değerin “Ahmet” içermediğini test edin

Assertions.assertFalse(donendeger.contains("Ahmet"))

- ✓ Örnek 3 : dönen değerin “Name:Ahmet” olduğunu test edin

Assertions.assertEquals(expected, actual)

Beraber Yapalım - Test senaryosu – 0

1. <https://www.ebay.com/> adresine git.
2. Açılan ekranında arama çubuğuuna Airpods yaz.
3. Enter'a basarak (submit ederek) aramayı yap.
4. Airpods araması yapıldığını doğrula.

Test senaryosu - 1

Test Adımları

1. <https://demoqa.com/login> adresine git.
2. Açılan ekranda UserName ve Password bölümlerini doldur.
3. Login butonuna bas.
4. Giriş yapılamadığını doğrula.

Test Verileri

UserName : newdemo

Password: Newdemotester@123

Örnek yapalım

1. https://the-internet.herokuapp.com/dynamic_controls adresine gidin.
2. Textbox'in etkin olmadigini(enabled) doğrulayın.
3. Enable butonuna tıklayın ve textbox etkin oluncaya kadar bekleyin.
4. "It's enabled!" mesajının goruntulendigini doğrulayın.
5. Textbox'in etkin oldugunu(enabled) doğrulayın

Test senaryosu - 2

Test Adımları

1. <https://demoqa.com/login> adresine git.
2. Açılan ekranınUserName ve Password alanlarını doldur.
3. Login butonuna bas.
4. Giriş yapıldığını doğrula

Test Verileri

UserName : newdemotester

Password: Newdemotester@123

Test senaryosu - 3

Test Adımları

1. <https://demoqa.com/text-box> adresine git.
2. Açılan ekranın Full Name, Email, Current Address, Permanent Address alanlarını doldur.
3. Submit butonuna bas.
4. Ekrana gelen text bölümündeki bilgilerin test verileri ile aynı olduğunu doğrula.

Test Verileri

Full Name: Ahmet

Email: ahmet@ben.com

Current Address : İstanbul

Permananet Address : İzmir

Beraber Yapalım

- ✓ https://automationexercise.com/test_cases adresindeki 4 numaralı test case'i beraber yapalım.

Örnek yapalım

- ✓ https://automationexercise.com/test_cases adresindeki 9 numaralı test case'i beraber yapalım.

Action Sınıfı

- ✓ Şimdi biz bu noktaya kadar driver'a elimiz ve gözümüz dedik. Yani elimizle gözümüzle yapacağımız her şeyi yapabilmesini istiyoruz.
- ✓ Ancak her zaman driver her zaman bize yetemeye biliyor.
- ✓ Peki ne zaman yetemez ?
- ✓ Test otomasyonu sırasında mouse ve klavye ile yapabileceğiniz tüm işlemleri yapmanıza imkan verir.
- ✓ Actions class method'larını mouse ve klavye method'ları olarak ikiye ayıralım.

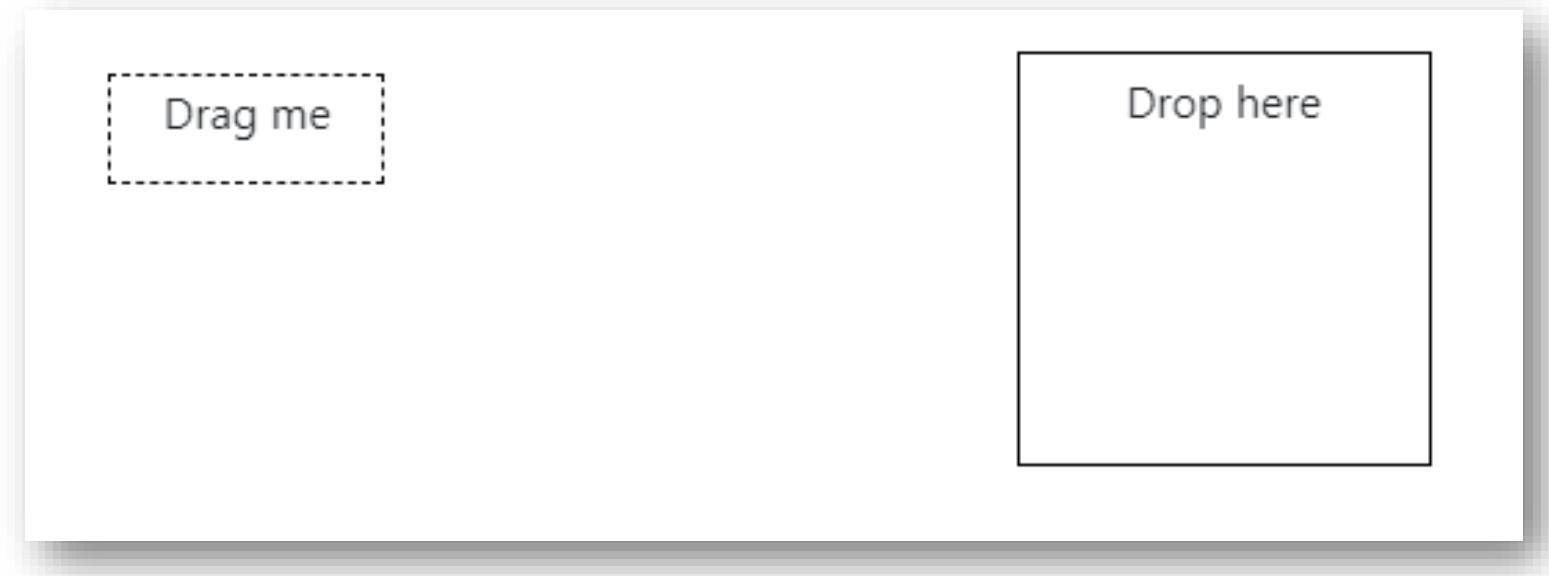
Mouse actions method'lari

Actions actions = new Actions(driver);

1. **actions.moveToElement()**: bir web elementi üzerine gidip beklememizi sağlar.
2. **actions.click()**: istediğimiz web element'e click yapar.
3. **actions.clickAndHold()**: istediğimiz web element'e basili bekler.
4. **actions.contextClick()**: istediğimiz web element'e sag click yapar.
5. **actions.doubleClick()**: istediğimiz web element'e çift click yapar.

Mouse actions method'ları

1. **actions.dragAndDrop()** : istediğimiz web element'i alıp, istediğimiz başka bir elementin üzerine bırakır.



Klavye actions method'ları

- ✓ Klavye'de her bir tuş için ayrı bir metod olusturmak mümkün değil.
- ✓ Temel olarak iki hareket var :
 - A. İstediğimiz tuşa tek seferlik basma
 - B. İstediğimiz tuşa istediğimiz sürece basılı tutma
 1. actions.sendKeys()
 2. actions.keyDown()
 3. actions.keyUp()

Test senaryosu

Test Adımları

1. https://automationexercise.com/test_cases adresine git.
2. Sayfanın urlini kaydet.
3. New Tab butonuna bas.
4. Açılan tabe git.
5. Tabde 'This is a sample page' yazısını doğrula.
6. Tab'i kapat ve ilk tabe dön.
7. 2. adımda kaydettiğin url ile bulunduğu tabın doğrulunu doğrula

Test senaryosu

Test Adımları

1. https://automationexercise.com/test_cases adresine git.
2. Sayfanın urlini kaydet.
3. New Window butonuna bas.
4. Açılan pencereye git.
5. Pencerede ‘This is a sample page’ yazısını doğrula.
6. Pencereyi kapat ve ilk Pencereye dön.
7. 2. adımda kaydettiğin url ile bulunduğu tabin doğrulunu doğrula

Örnek yapalım

- ✓ https://automationexercise.com/test_cases adresindeki 1 numaralı test case'i beraber inceleyelim ve yapalım.

Selenium ile Tabler arası navigasyon

- ✓ Öncelikle WebDriver pencereler ve tabler arası bir ayırm yapmaz. Eğer bulunduğuuz sayfada yaptığınız bir işlem yeni bir pencere veya tab açarsa selenium bu pencerede veya tabde çalışmaya devam eder.
- ✓ Tabler yada sayfalar arası çalışırken bilmeniz gereken şey her tabin ve pencerenin kendine ait özgün bir tanımlayıcısı bulunur. Bu tanımlayıcıya '**driver.getWindowHandle();**' metodu ile ulaşabilirsiniz.
- ✓ Bu tanımlayıcı kod yazarak tabler veya pencereler arası geçisi sağlamak için önemlidir.

Selenium ile Tabler arası navigasyon

```
//Öncelikle bulunduğuuz ana panelin idsini kaydedelim.
```

```
String originalWindow = driver.getWindowHandle();
```

```
//Açık Başka bir tab olup olmadığını kontrol edelim.
```

```
assert driver.getWindowHandles().size() == 1;
```

```
//Yeni bir pencerede açılacak olan linke tıklayalım.
```

```
driver.findElement(By.linkText("new window")).click();
```

```
//Eğerki Boş yeni bir tab açmak istiyorsak.
```

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

```
//Yeni pencere açılana kadar bekleyelim
```

```
wait.until(numberOfWindowsToBe(2));
```

Selenium ile Tabler arası navigasyon

```
// Yeni Pencereyi bulana kadar pencereler arası gez. Bulunca o pencereye geçiş yap.  
for (String windowHandle : driver.getWindowHandles()) {  
    if(!originalWindow.contentEquals(windowHandle)) {  
        driver.switchTo().window(windowHandle);  
        break;  
    }  
}  
//Yeni Pencere yüklenene kadar bekle  
wait.until(titleIs("Selenium documentation"));
```

- ✓ **DipNot:** Selenium 4 ile birlikte bu işlemi kısalta bilmek adına yeni bir metod yayınladı. ([NewWindow](#))

Tabi kapatmak

```
//Bulduğumuz pencereyi kapatır.  
driver.close();
```

```
//Tekrar eski pencereye geçiş yapmalıyız.  
driver.switchTo().window(originalWindow);
```

- ✓ Eğer switchTo ile geçiş yapmayı unutursak testimiz devamında kapalı pençede işlem yapmaya çalışacağı için hata verecektir.

✓ <https://demoqa.com/browser-windows> sayfasına gidin.

Teşekkürler...