Tiberium Keyword Driven Test Automation Framework

Keyword Driven Testing ve Behavior Driven Development Yaklaşımı

Burak AVCI
Test Automation Application Lead
www.burakavci.com.tr

Tiberium Framework

Tiberium; Unit ve Regresyon Testlerinizi Kelime Güdümlü olarak bilinen Keyword Driven Testing mantığı ile hazır metotlar yani Keyword'ler ile yazmanızı amaçlar. Behavior Driven Development yaklaşımı ile aynı olup KDT özelliği de eklenmiştir.

- KDT - Keyword Driven Test Automation Framework
- Written in Visual C# / .NET (Advanced OOP)
- BDD - Behavior Driven Development Yaklaşımı
- **Platforms:** Web Applications
- **Operating System:** Cross-Platform
- **Underlying Framework:** Selenium WebDriver
- **Library:** Selenium.WebDriver & Selenium.Support
- Framework dosyası içindeki **Drivers** klasörünü Local Bilgisayardaki **C:** dizini içine atınız.
- Test Scripti oluştururken **Silk Selenium WebDriver** ile önce Record ederek Test iskeletini çıkarabilirsiniz.

- **Version:** 3.5 | **Github:** https://github.com/burakavcioglu/Tiberium

Örnek Proje Yapısı

Proje yapısı oluştururken **ExampleProject**'i örnek alabilirsiniz.

Yandaki Resimde de görüldüğü gibi ilk olarak **ProjectDriver** adında bir projemiz için Base Class oluşturuyoruz. Ve bu sınıfı Tiberium'a kalıtım vererek Framework ile bağlıyoruz.

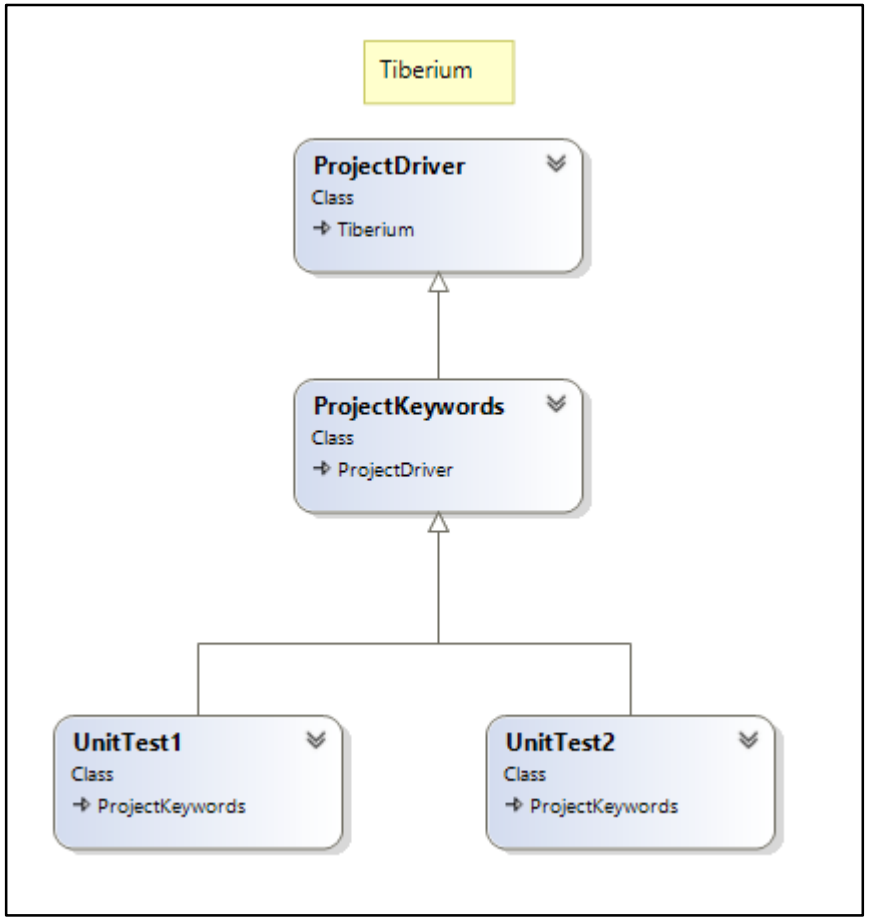
public class ProjectDriver : Tiberium

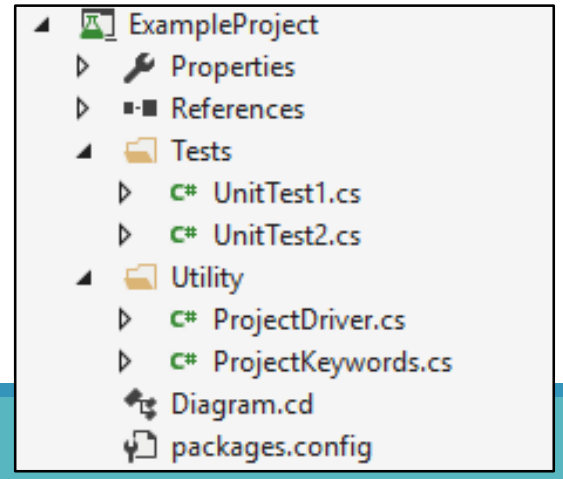
Driver Class'ı içine projeye ait Custom Driver ve Static Tanımları yazmak için kullanabilirsiniz.

ProjectKeywords Class'ı içine projeye ait Custom Keyword yazmak için kullanabilirsiniz.

Testleri oluşturmak içinse Add **Unit Test** diyerek oluşturabilir ve kelime güdümlü testlerinizi yazmaya başlayabilirsiniz.

Sabit Driver değişkeni **Device** olarak tanımlıdır.

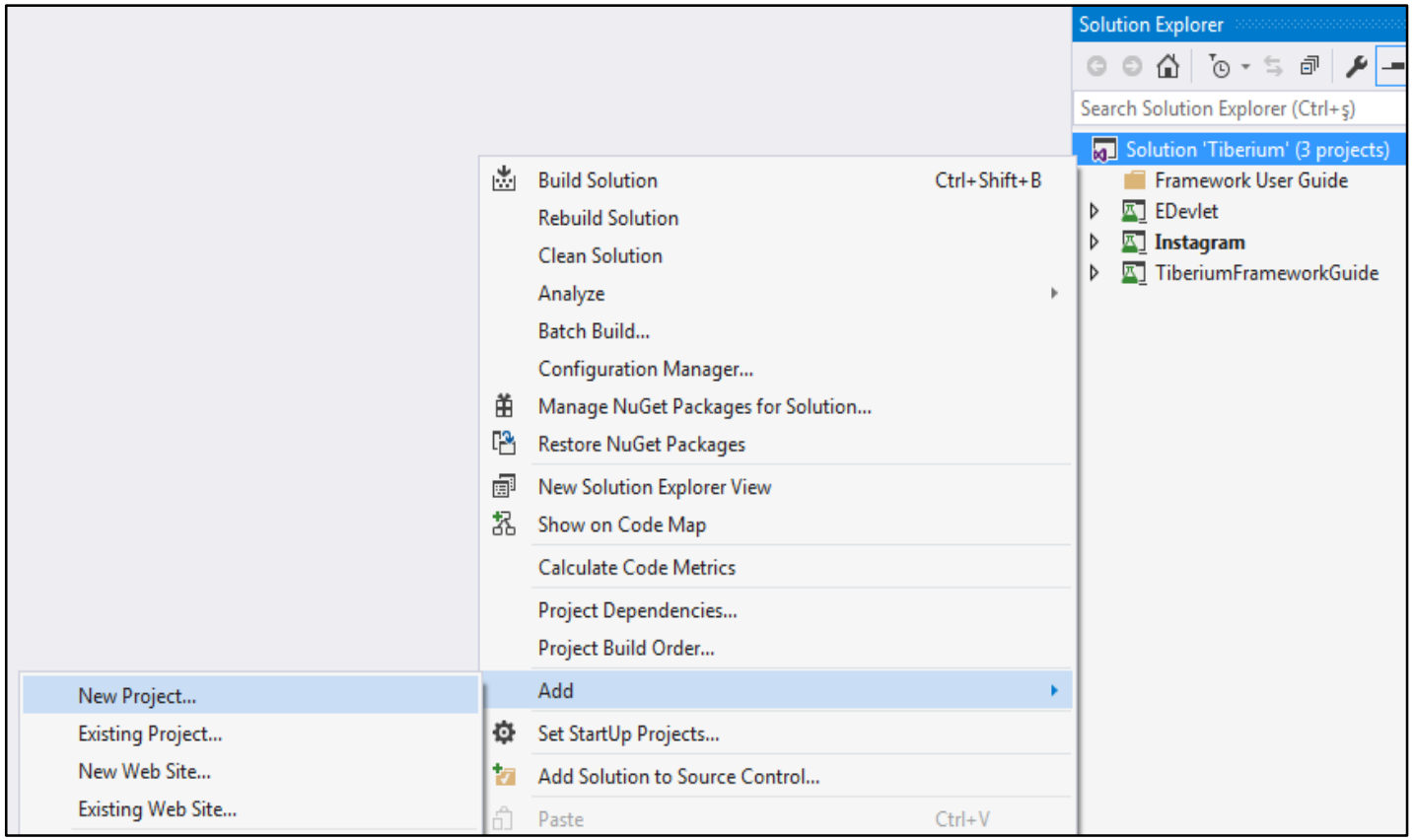




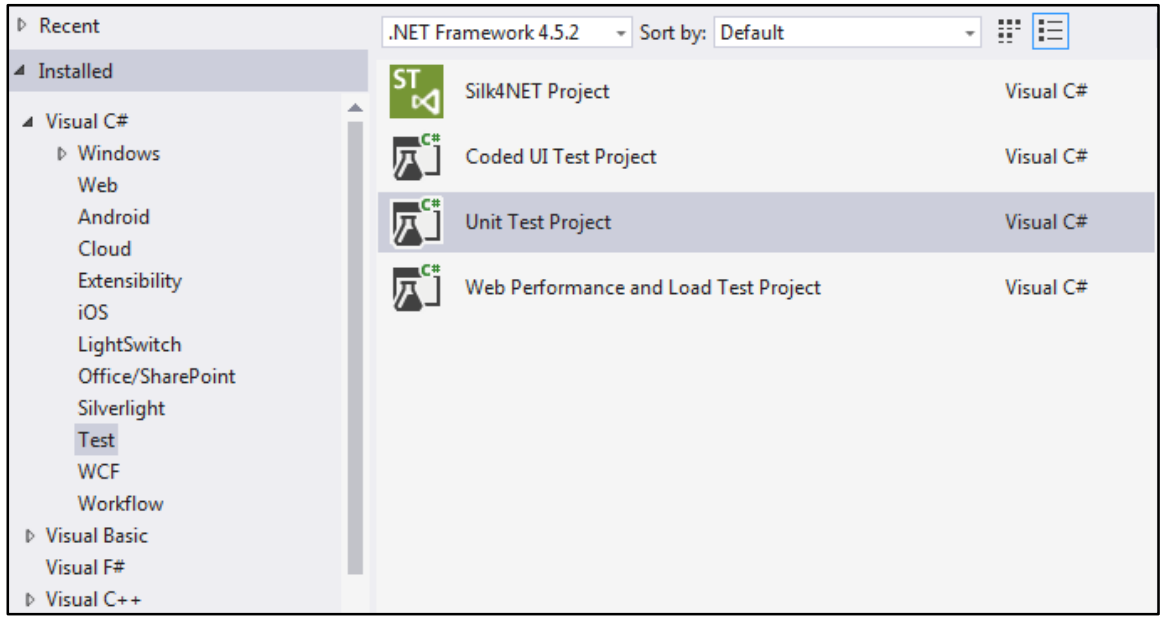
Yeni Bir Proje Oluşturmak

Microsoft Visual Studio'da Selenium Test Projesi Oluşturmak

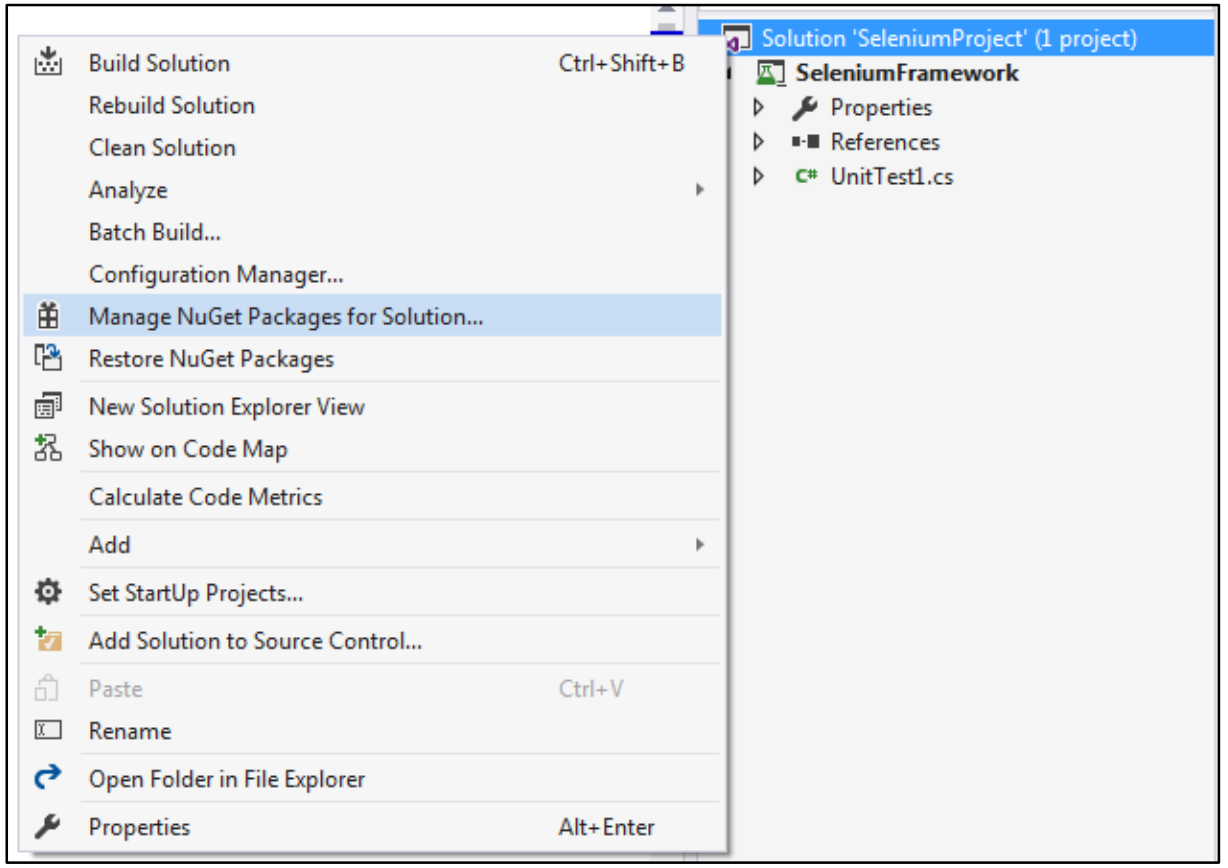
Testini yazacağınız Test Projesini aşağıdaki gibi oluşturabilirsiniz. Ben Örnek olarak E-Devlet ve birkaç Projenin testlerini ekledim. Buradaki örnek projelere bakarak mantığı anlayabilirsiniz.



İlk olarak Tiberium Solution üzerine gelip Sağ tıklayın. **Add** kısmından **New Project** sekmesine geliniz.



Test sekmesinden **Unit Test Project** seçiniz ve Projenin adını yazıp OK demeniz yeterli olacaktır.

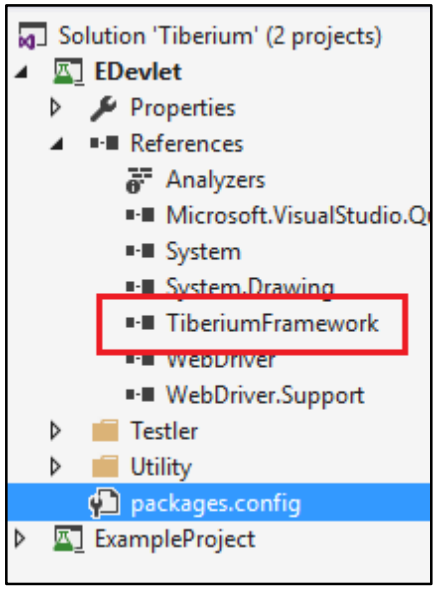


Manage NuGet'den Selenium ve Üzerinde çalışmak istediğiniz Framework'ün DLL'lerini Tüm Projelere Install diyerek ekleyebilirsiniz. Tiberium Keyword'lerini kullanmak için **Tiberium** Klasörü içindeki **TiberiumFramework.dll** dosyasını **References** kısmından eklemeniz yeterli olacaktır.

Ayrıca **Selenium.WebDriver** ve **Selenium.Support** paketlerinide NuGet'den çekebilirsiniz.

Piyasaki Diğer Framework'ler ve DLL'ler

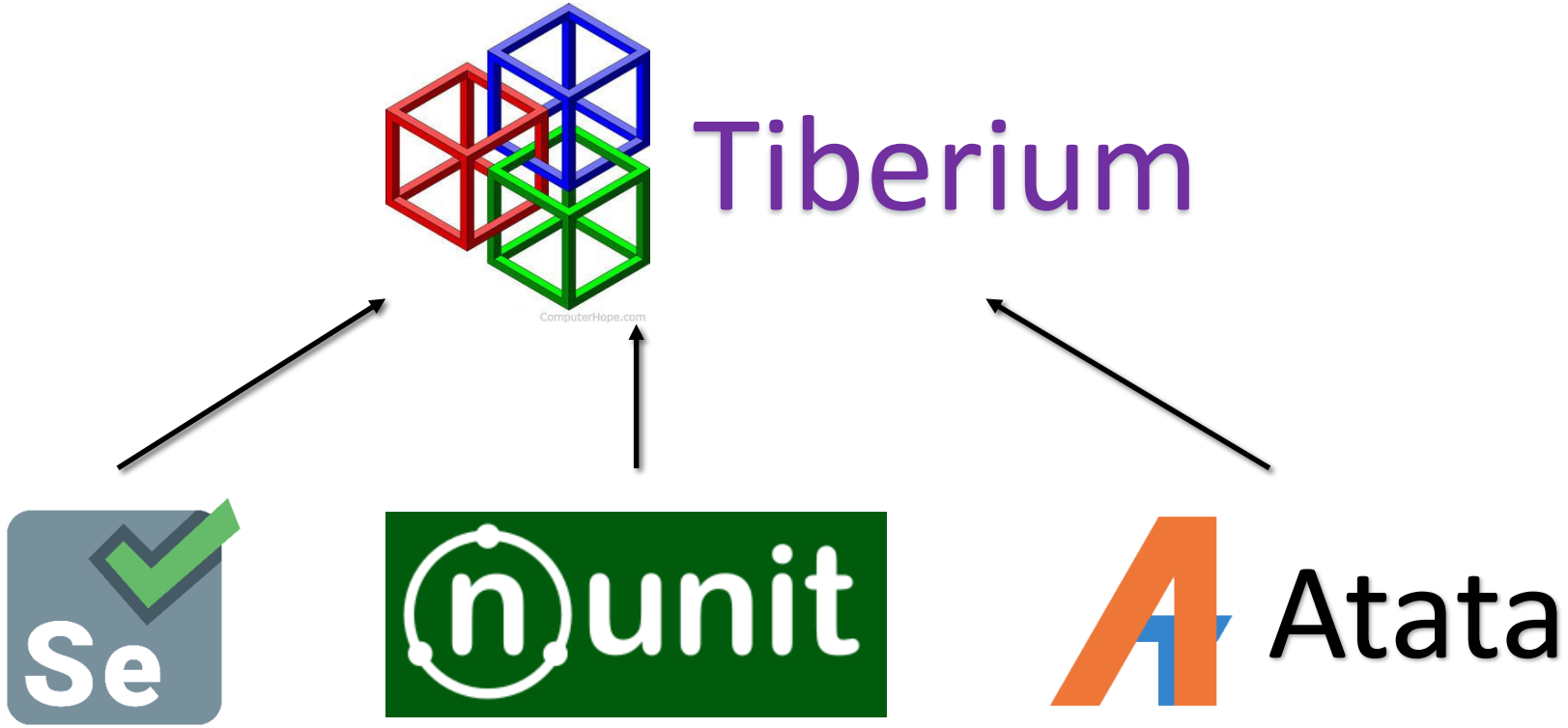
- NUnit (Unit Testing Framework TDD)
- NUnit3TestAdapter,
- NMock
- Atata Framework (Selenium Framework)
- Atata.WebDriverExtras
- ThoughtWorks.Selenium.Core
- ThoughtWorks.Selenium.IntegrationTests
- ThoughtWorks.Selenium.UnitTests



Mimari Altyapı - Architectural Infrastructure

Alt yapı olarak Selenium Web Otomasyon Framework kullanılmıştır ve **Selenium.WebDriver & Selenium.Support** Libraries eklenmiştir.

Visual C# ile yazılıp bir Class Mimarisi olduğundan alt yapıda başka Framework'ler kullanıp sizde kendi Framework alt yapınızı oluşturabilir ve bunu Kelime Güdümlü KDT mantığı ile orta katmanı hazırlayabilirsiniz.



Keywords ve Açıklamaları

Unit, Regresyon Testlerini yazmak için kullanacağınız Keyword'ler

Genel Keyword'ler

- **DegerGir**(string aciklama, string text, string value, string ElementType)

Herhangi bir alana INPUTİ TEXTArea veya diğerleri, değer girmeye yarar.

- **ListedenSec**(string aciklama, string value, string deger, string ElementType)

- **LokasyonFind**(string value, string filepath, string ElementType)

- **Temizle**(string value, string ElementType)

Değer girilmiş bir alanın içindeki değeri temizler, Clear mantığı ile çalışır.

- **MouseMove**(string value, string ElementType)

Click-Tıklama İşlemleri

- **Tikla**(string value, string ElementType)
- **FazlaTikla**(string value, int tiklama, string aciklama, string ElementType)
- **BekleTikla**(string value)
- **ContextClick**(string value, string ElementType)
- **DoubleClick**(string value, string ElementType)
- **LokasyonClick**(string value, string ElementType, int X, int Y)

URL-Link İşlemleri

- **ChromeURLOpen**(string url)
- **FirefoxURLOpen**(string url)
- **PageRefresh**() = Sayfayı yeniler.

- **PageBack**()

Bir önceki sayfaya geri döner.

- **OpenLink**(string browser, string url)

Tarayıcı belirlenir (Chrome, IE, Firefox) ve gidilecek Link yazılır.

Senkronizasyon İşlemleri

- **Kontrol**(string value, string ElementType)

Ekrandaki Objeyi (XPath, Metin, ID) Kontrol eder, Obje varsa devam eder yoksa Failed verir.

- **Bekle**(int saniye)

Verilen süre boyunca bekler saniye cinsinden.

- **TestBitti**()

Test Steplerinin sonuna eklenir, Tarayıcıyı kapatır.

- **EkranMaximize**()

Browser ekranını Maximize eder.

- **AllBrowserKill**()

Tüm açık tarayıcıları kapatır. Herhangi bir parametre almaz.

Örnek Testler

```
[TestClass]
0 references
public class MevduatFaizHesaplama : EnparaKeywords
{
    [TestMethod]
    0 references
    public void MevduatFaizHesaplamaTest()
    {
        // QNB Finansbank Enpara Mevduat Faiz Hesaplama Testi
        // Keyword Driven Testing - KDT

        // Behavior Driven Development - BDD Yaklaşımı

        AllBrowserKill();
        OpenLink("Chrome", "https://www.qnbfinansbank.enpara.com");
        Tikla("ORANLAR VE FİYATLAR", Metin);
        Tikla(MevduatOranlarLogo, XPath);
        EnparaHesapTutarGir("50000"); //Enpara Special Keyword
        ListedenSec("Para Birimi", "drpCurrency", "TL", ID);
        Tikla("rdHesapVadeli", ID); //Vadeli Hesap Radio Button
        ListedenSec("Toplam Mevduat Büyüklüğü", "drpAumRange", "0 - 100.000 TL arası", ID);
        ListedenSec("Vade", "drpVade", "32 gün", ID);
        Tikla("mevduatCalculation", ID); //Hesapla
        Kontrol("vadeSonuBakiye", ID);
        Bekle(3); //Hızlı Geçtiği için bekletiyorum.
        TestBitti();
    }
}
```

```
[TestClass]
0 references
public class SicilKayitSorgulama : EDevletLayer
{
    [TestMethod]
    | 0 references
    public void AdliSicilKayitSorgulama()
    {
        //BDD - Behavior Driven Development Test Automation

        //Adli Sicil Kaydı Sorgulama Testi

        EDevletLogin("Chrome", TCKimlik, Sifre); //E-Devlet Login Keyword: Tarayıcı, TCKN, Şifre
        Tikla(EHizmetlerLocator, XPath);
        Tikla(Adalet, XPath);
        Tikla(AdliSicilSorgu, XPath); //AdliSicilSorgu XPath - "Adli Sicil Kaydı Sorgulama" Metin
        Tikla("chkOnay", Name);
        Tikla("btn", Name);
        ListedenSec("Kurum Türü", "cmbKurumTuru", "Özel", ID);
        ListedenSec("Belgenin Neden Verileceği", "belgeNedenVerilecek", "Ehliyet", ID);
        DegerGir("Belgenin Nereye Verileceği", "TIBERIUM", "belgeNereyeVerilecek", ID);
        Tikla("btn", Name);
        Kontrol(KontrolSicil, XPath); //Bu sayfayı DOĞRUDAN YAZDIRMAYINIZ! Metni Kontrol Edilir
        Tikla("Kayıtı Belgelerim", Metin);
        Tikla("Geri Dön", Metin);
        TestBitti();
    }
}
```

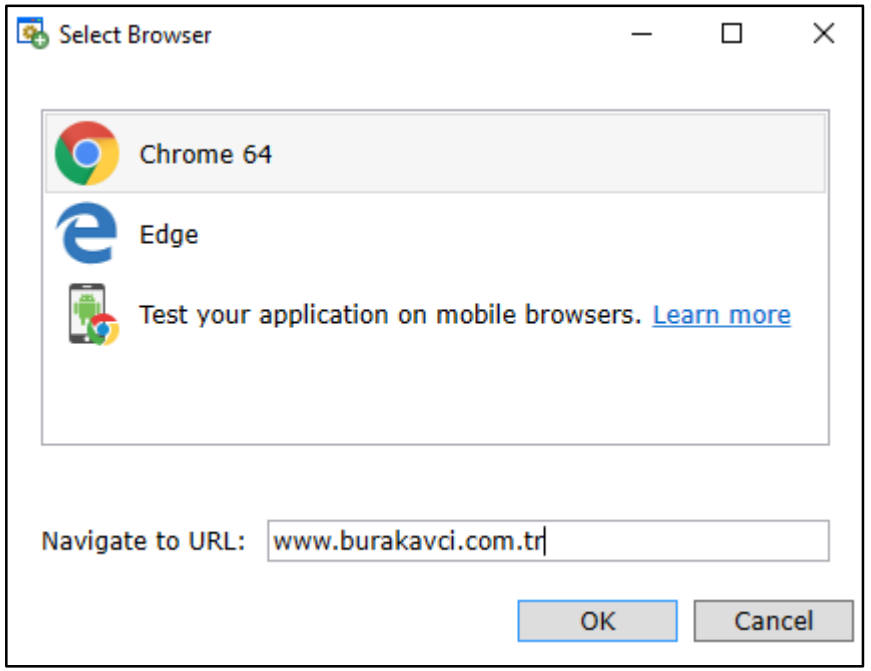
SILK Selenium WebDriver

Micro Focus'un Ücretsiz (FREE) ürünü olan SILK Selenium WebDriver ile Selenium alt yapılı Test Scriptleri yazabilirsiniz. Oluşturduğunuz Testleri istediğiniz programlama dilinde Export ederek çıktısını alabilirsiniz.

SILK WebDriver ürününü https://www.microfocus.com/products/silk-portfolio/silk-webdriver adresinden Download edebilirsiniz. Kurulumu yaptıktan sonra Mail adresiniz ile üye olmayı unutmayınız.

Kullanımdan bahsedecek olursak; Uygulamayı açınca **Record new script** diyerek bir Web sitesi üzerinde Record işlemi yaparak test Scriptlerinizi yazabilir ve test senaryonuzu oluşturabilirsiniz.

Browser olarak Google Chrome veya Firefox seçip **Navigate to URL** kısmına Adresi yazıp OK demeniz yeterlidir.



SILK Selenium WebDriver Faydaları

- Selenium metotlarını kullanarak Web üzerindeki senaryolarınızı **Record** yaparak istediğiniz programlama dilinde Test sınıfı olarak kayıt edebilirsiniz, daha sonra bunları Manuel eklemeler yaparak geliştirebilirsiniz.

- Objenin Xpath, ID, Text ve diğer değerlerini **SWD** formatındaki dosya içinden alabilir ve Test Scripti yazarken buradaki bilgilerden yararlanabilirsiniz.

- Birden fazla **OOP** dilini desteklemektedir.

- Test etmek istediğiniz senaryonun **Test İskeletini** Record yaparak oluşturup daha sonra bu Script üzerine Manuel metotlar ekleyebilirsiniz. Bu sayede Tüm senaryo steplerini yazmaktan kurtulur, hemde Framework'ün o nesne için nasıl davrandığını daha iyi anlayabilirsiniz.

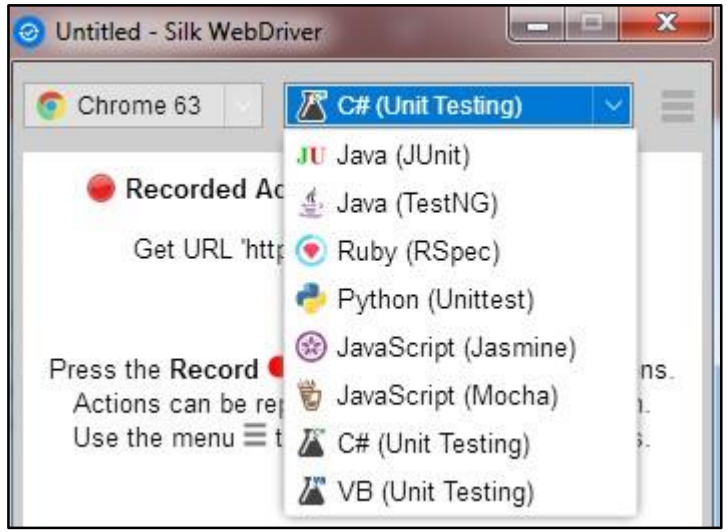
- Google Chrome ve Firefox Web tarayıcıları üzerinden Record işlemi yaparak Test Script dosyalarınızı oluşturabilirsiniz.

- Selenium ile otomasyon yazan Tüm Developer'lara önerebileceğim **ücretsiz** bir araçtır.

Test Scriptinizi oluşturduktan sonra arka planda oluşan otomatik kodu hangi dilde almak istediğinizi seçebilirsiniz. Record işlemi bittiğinden Stop diyerek SWD uzantılı Record dosyanızı kaydedebilir ve her defasında bu senaryonuzu otomatik koşabilirsiniz.

- **Java JUnit**
- **Java TestNG**
- **Ruby Rspec**
- **Python Unittest**
- **JavaScript Jasmine**
- **C# Unit Testing**
- **VB Unit Testing**

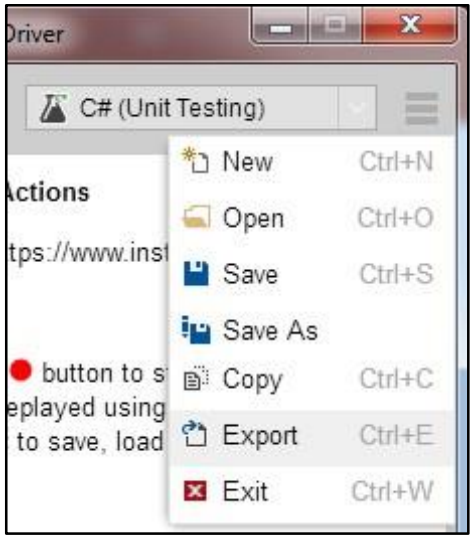
Dillerini destekleyip bunlardan birini seçip Export ederek Test Script kodunu alabilirsiniz, Selenium kullanan Developer'lar en çok aracın yararını bu kısımda görecektir. Ben C# Unit Testing kullanıyorum.

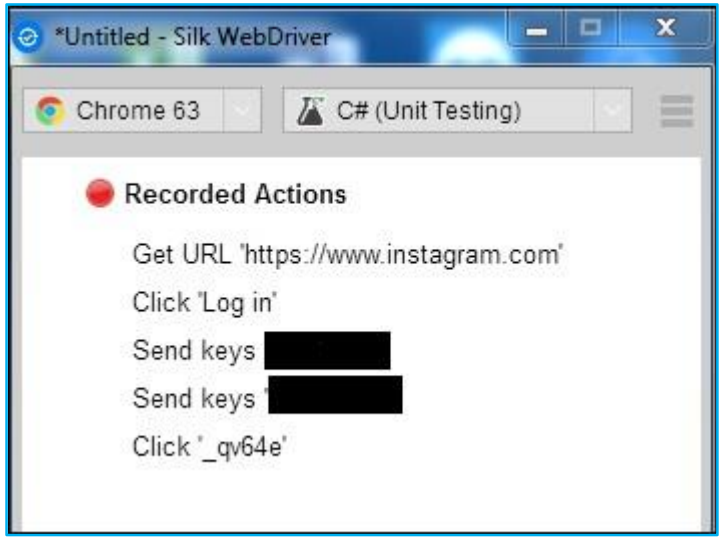


```
using System;
using OpenQA.Selenium;
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace 1
{
[TestClass]
public class 1
{
    IWebDriver driver;

    [TestInitialize]
    public void Setup()
    {
        // Add the driver directory "C:\Program Files (x86)\Si
        // uncomment the following line
        // driver = new OpenQA.Selenium.Chrome.ChromeDriver();
        // and delete the next line
        driver = new OpenQA.Selenium.Chrome.ChromeDriver(@"C:\
        driver.Manage().Timeouts().ImplicitWait = TimeSpan.Fro
    }

    [TestMethod]
    public void TestMethod()
    {
        driver.Url = "http://www.burakavci.com.tr";
        driver.FindElement(By.LinkText("Port 80")).Click();
        driver.FindElement(By.LinkText("Coding")).Click();
    }

    [TestCleanup]
    public void TearDown()
    {
        if (driver != null)
        {
            driver.Dispose();
        }
```

Burada üzerine değinmek istediğim konu; Siz Silk Test Framework'ünü kullanıp Selenium Framework'ünü de içine entegre ettiğinizde, Unit Test yazan Test otomasyon mühendisleri bu araçla Unit Testlerini yazabilir ve Framework
içerisinden Selenium ile de çalışmış olur. Ayrıca başka bir Tool kullanmasına da gerek kalmaz.

Sadece Selenium mimarisi ile çalışanlar da bu aracı kullanarak Web otomasyon geliştirmelerini daha kaliteli hale getirebilir.

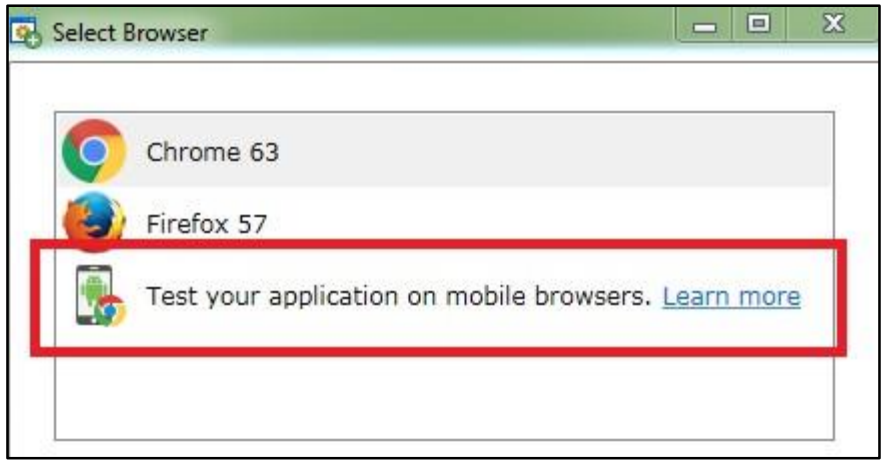




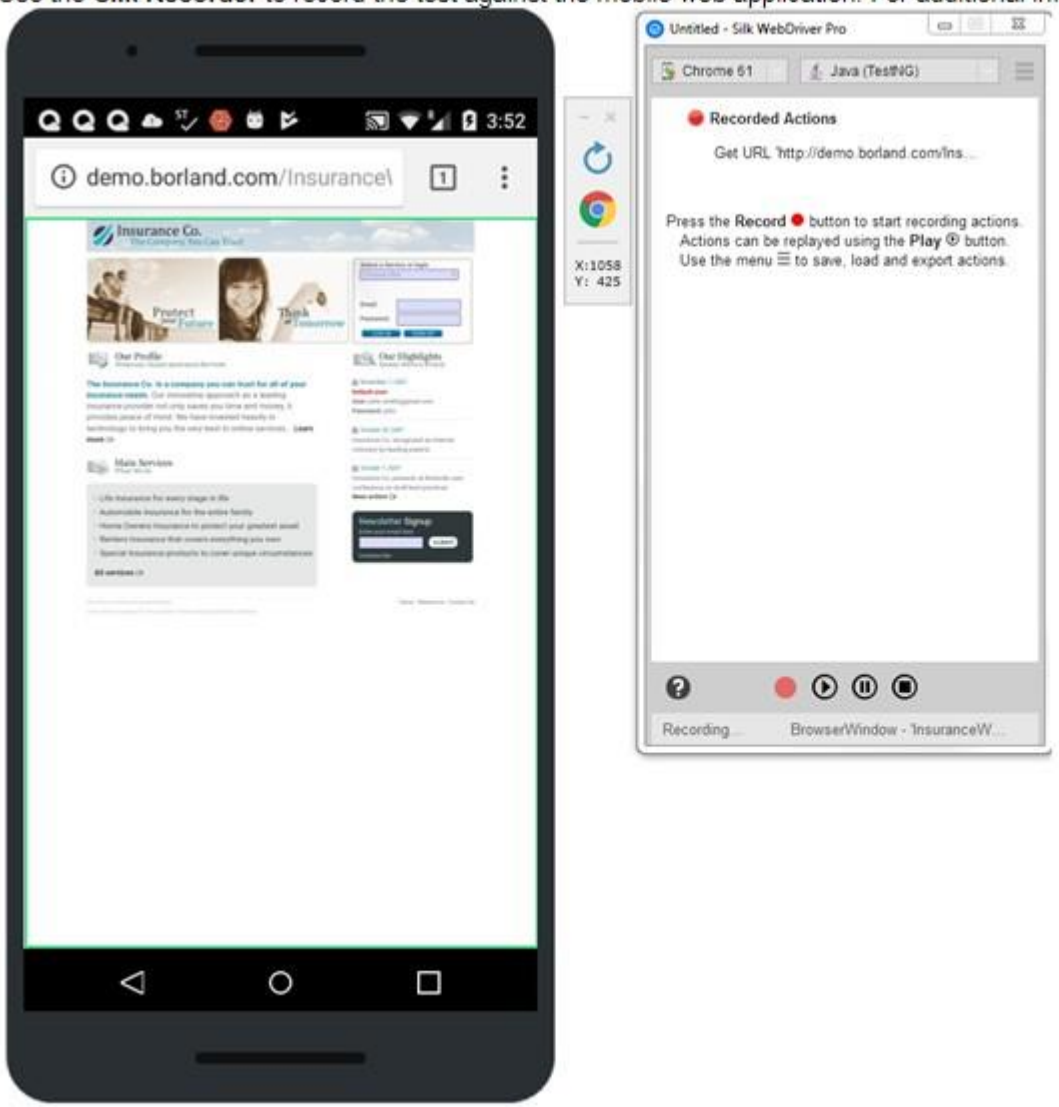
```
IWebDriver driver;

[TestInitialize]
public void Setup()
{
    // Add the driver directory "C:\Program Files (x86)\Silk\Silk WebDri
    // uncomment the following line
    // driver = new OpenQA.Selenium.Chrome.ChromeDriver();
    // and delete the next line
    driver = new OpenQA.Selenium.Chrome.ChromeDriver(@"C:\Program Files
    driver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(5);
}

[TestMethod]
public void TestMethod()
{
    driver.Url = "https://www.instagram.com";
    driver.FindElement(By.LinkText("Log in")).Click();
    driver.FindElement(By.Id("f1d8e14bafea3e4")).SendKeys("USER");
    driver.FindElement(By.Id("fb36d81b8c48e")).SendKeys("PASS");
    driver.FindElement(By.ClassName("_qv64e")).Click();
}
```


```
    "commonClass" : "WebElement",
    "actionName" : "Click",
    "parameters" : { },
    "locatorInfos" : [ {
      "className" : "WebElement",
      "isDirectChild" : false,
      "attributes" : {
        "linkText" : "Log in",
        "xpath" : "//a[text()=\"Log in\"]"
      }
    } ]
  }, {
    "commonClass" : "WebElement",
    "actionName" : "SendKeys",
    "parameters" : {
      "keys" : "USERNAME"
    },
    "locatorInfos" : [ {
      "className" : "WebElement",
      "isDirectChild" : false,
      "attributes" : {
        "id" : "f1d8e14bafea3e4",
        "name" : "username",
        "xpath" : "//input[@id=\"f1d8e14bafea3e4\"]"
```

Yukarıdaki gibi sağ tarafta Visual C# kodu ile oluşturulan Script ve Sağ tarafta SWD formatında Test Script'i. SWD dosyası içinde her nesnenin Xpath ve diğer bilgilerini otomatik çekerek kaydeder. Siz Manuel müdahale etmek istediğiniz zaman buradaki bilgileri kullanabilirsiniz. Örneğin **LinkText** verildiği gibi çalışmazsa **Xpath** bilgisini alarak deneyebilirsiniz.



Use the **Silk Recorder** to record the test against the mobile web application. For additional information, see Recording Tests.



Aynı Zamanda Web Sitesini Mobil Tarayıcıda da Record ederek koşturabilirsiniz.

Bu özellik Mobil Uygulama(Mobile App) ile karıştırılmamalıdır. Sadece **Android Browser** üzerinde açılan siteyi Record edebilirsiniz.

Kullanabilmek için Android SDK kurulumu yapmanız gerekir.


MICRO FOCUS
COBOL
SERENA
Borland
Attachmate
Novell
SUSE
Progress
NetIQ
40 years
Hewlett Packard Enterprise
Network Management/ Data Protector
FORTIFY
PSWARE
MERCURY
ArcSight
VERTICA
Peregrine
SHUNRA
ATALLA
Voltage
30 years