WebDriver.io

Kurulum Kılavuzu

|  |  |
| --- | --- |
| Doküman Adı | KK-WebDriver.io-230112 |
| Doküman Dosya Adı | KK-WebDriver.io-230112-v2.0. |
| Oluşturulma Tarihi | 12.01.2023 |
| Hazırlayan | Çağdanur Öten |
| Kontrol Eden |  |
| Versiyon | V1.0. |
| Yayınlama Tarihi |  |

İçindekiler

[1. WebDriver.io Kullanım Amacı? 4](#_Toc120884716)

[2. WebDriver.io Kurulumu 4](#_Toc120884717)

[3. WebDriver.io Test Scriptleri 7](#_Toc120884718)

[4. Allure Reporter Kullanım Amacı? 12](#_Toc120884719)

[5. Allure Reporter Kurulumu 1](#_Toc510186269)2

## WebDriver.io Kullanım Amacı?

WebDriver.io geliştirme yapılmış projelere ait testlerin yapılmasını sağlayan yapıdır. Farklı senaryoları koda dökerek otomasyonlar yazılmasını sağlar. Bu sayede geliştirilen projeleri manuel yerine otomasyonlar ile test ederek gözden kaçırmalara ya da insani hatalar yapmaya engel olabiliriz.

## WebDriver.io Kurulumu

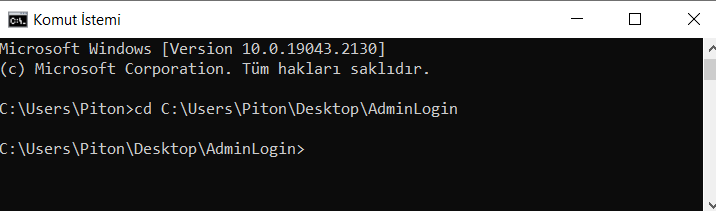
Masaüstünde bir klasör oluşturulur.

Arama tuşuna cmd yazılır ve terminal ekranı açılır.

“cd” komutuyla oluşturulan klasöre gidilir.

cd C:\Users\Piton\Desktop\AdminLogin

Metin Kutusu 1 Klasöre Erişme



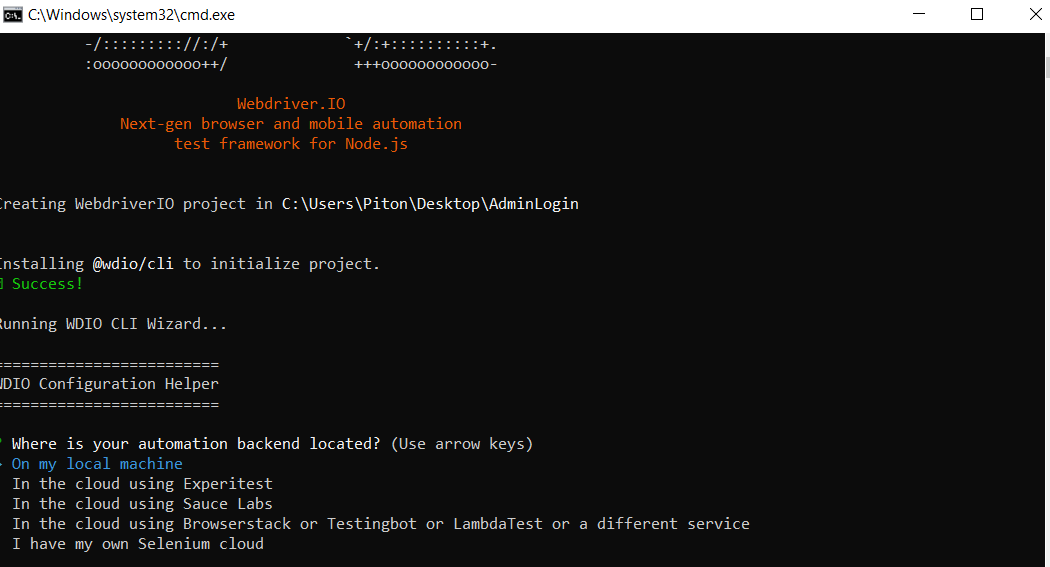
Şekil 1 Terminal ekranında klasöre erişme

\*\*\*Not: Bu doküman içerisinde terminal(cmd) ekranında yazılan tüm kodlar klasör içerisinde yazılır

Aşağıdaki kod ile kurulum için gerekli dosyalar otomatik olarak oluşturulur.

npm init wdio .

Metin Kutusu 2 WebDriver.io Kurulum



Şekil 2 Kurulum ekranı

Gerekli ayarlamar aşağıdaki şekilde düzenlenir. Which reporter do you want to use? adımında spec şeçili halde iken allure değildir. Yön tuşları ile aşağı inerek space tuşu ile seçim yapılır ve enterlanır.

A project was detected at "C:\Users\Piton\Desktop\Yeni klasör (2)", correct? **Yes**

Where should your tests be launched? **local - for e2e testing of web and mobile applications**

Where is your automation backend located? **On my local machine**

Which framework do you want to use? **Mocha (https://mochajs.org/)**

Do you want to use a compiler? **No!**

Do you want WebdriverIO to autogenerate some test files? **Yes**

Where should these files be located**? C:\Users\Piton\Desktop\Yeni klasör (2)\test\specs\\*\*\\*.js**

Do you want to use page objects (https://martinfowler.com/bliki/PageObject.html)? **Yes**

Where are your page objects located? **C:\Users\Piton\Desktop\Yeni klasör (2)\test\pageobjects\\*\*\\*.js**

Which reporter do you want to use? **spec, allure**

Do you want to add a plugin to your test setup? **Yes**

Do you want to add a service to your test setup? **chromedriver**

What is the base url? (http://localhost) **What is the base url?** [**https:www.google.com**](https://www.google.com/)

Do you want me to run `npm install` **Yes**

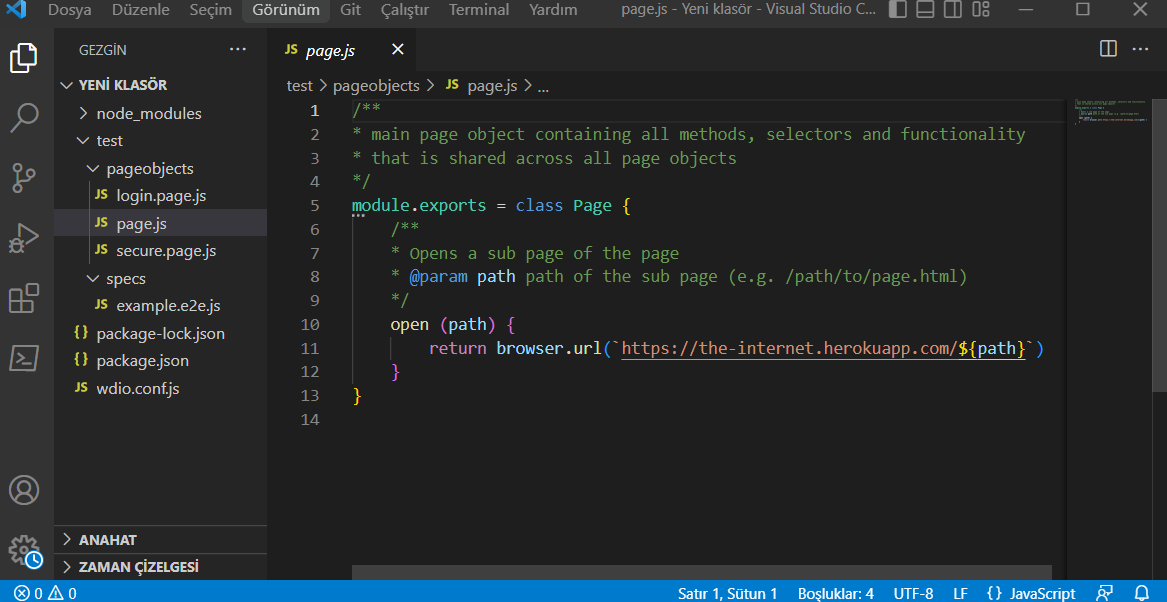
Metin Kutusu 3 Konfigürasyonların yapılması

WebDriver.io kurulumu başta oluşturulan klasör içerisinde tamamlandı. WebDriver.io içerisinde hazır bir Login kodu ile gelir. Bu sayede hali hazırda bulunulan cmd ekranında “npx wdio” komutu ile hazır kod çalıştırılabilir.

Geliştirme yapabilmek için bir IDE yüklenmeli ve bu klasör IDE içerisinde açılmalıdır. Ardından hazır kod üzerinde düzenlemeler yapılarak istenilen proje test edilebilir. En yaygın kullanılan IDE Visual Studio Code’dur. Bu dökümanda da Visual Studio Code kullanılacaktır. Yine bulunulan cmd ekranında “code .” koduyla klasör visual studio içerisinde açılabilir.

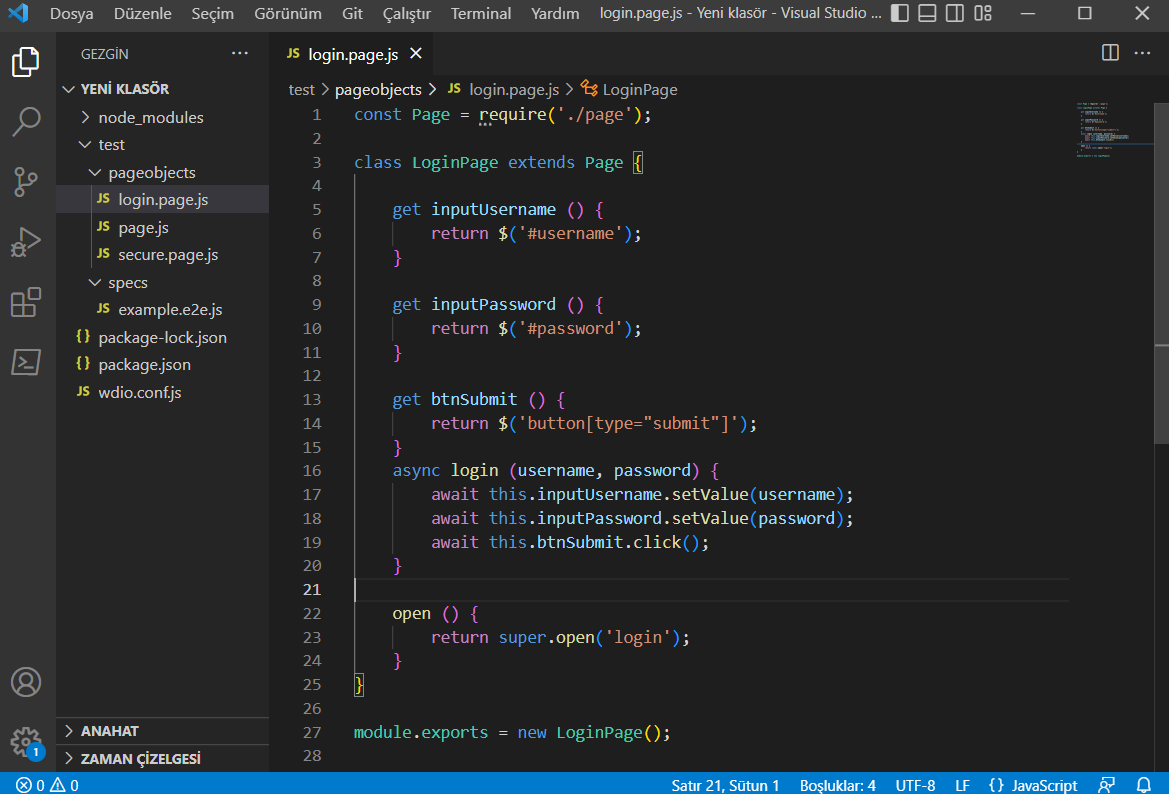
## WebDriver.io Test Scriptleri

Example.e2e.js projesi Visual Studio’da açılır ve pageobjects klasörünün altında page dosyasına gidilir. Burada test edilen örnek projenin URL adresinin tanımlandığı görülür.



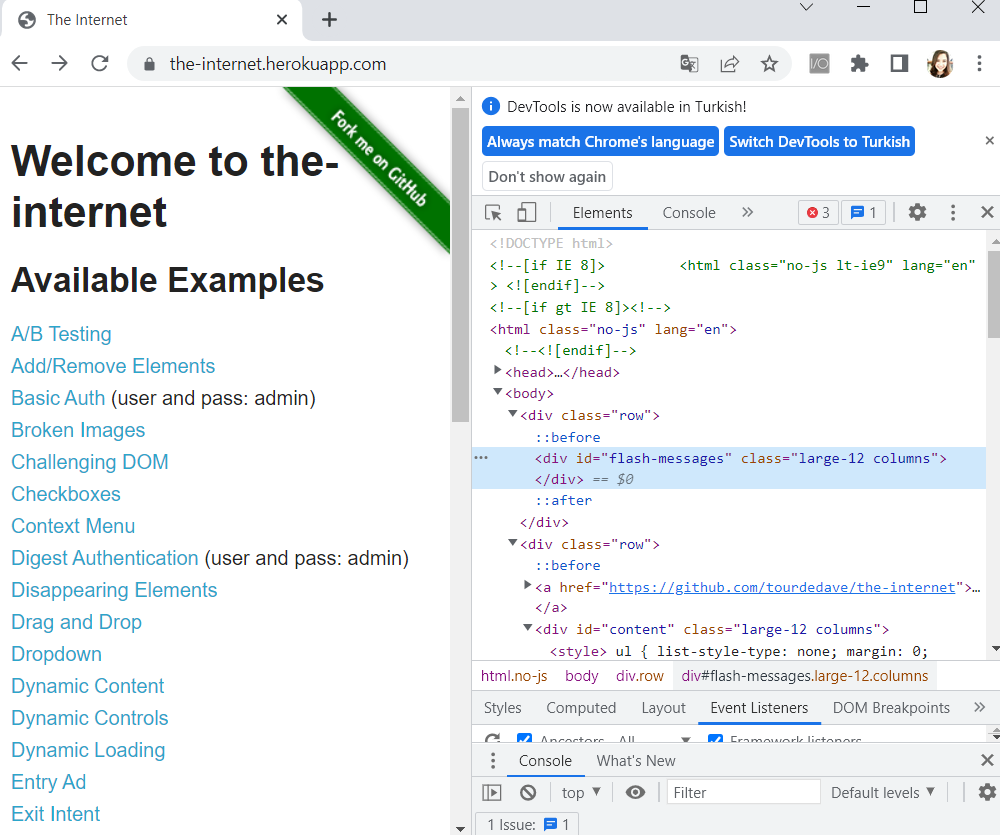
*Şekil 3 Page Sayfası*

Login page sayfasına gidilir ve orada test esnasında kullanılacak elementler get() methodu ile tanımlanır, web sitesindeki adresleri belirtilir.



*Şekil 3 Element Tanımlama*

**Id:** Adresler id türünde çekilir ise $(‘#flash-messages’) şeklinde yazılması yeterlidir.*(Bakınız Şekil4)*



*Şekil 4 Adresin id ile çekilmesi*

**Xpath:** Adresler xpath yoluyla çekilir ise $(‘..xpath..’) şeklinde resimdeki button tanımlama kısmında göründüğü gibi direkt yazılması yeterlidir.*(Bakınız Şekil3)*

Tanımlamalar bittikten sonra aynı dosya içerisinde tanımlamaların altında fonksiyonlar oluşturulur. Elementler üzerinde yapılmak istenilen işlemler belirlenir. *(Bakınız Şekil 3)*

await this.journeyTempl.waitForExist({timeout: 5000})

*Şekil 6 Timeout Kod Satırı*

journeyTempl elementi görünür hale gelene kadar 5 sn bekle ve öyle devam et anlamına gelir.

await this.journeyTempl.click();

*Şekil 7 Click Methodu*

journeyTempl elementine tıklamak için click() methodu kullanılır.

await browser.pause(3000);

*Şekil 8 Pause*

Çok hızlı çalıştığı için bazı işlemler takip edilemeyebilir. Bu noktada 3 sn browser’ı durdur diyerek işlem izlenebilir.

console.log(await this.titleCheck.getText());

*Şekil 9 GetText Methodu*

titleCheck elementindeki değer getText() methodu ile alınır ve konsol ekranına yazdırılır.

await this.inputPassword.setValue(password);

*Şekil 10 SetValue Methodu*

password parametresindeki değer inputPassword elementine setValue() methodu ile gönderilir.

**Seçilen element button ise;**

*Click();* ile tıklama methodu çalıştırılır.

**Seçilen element TextBox ise;**

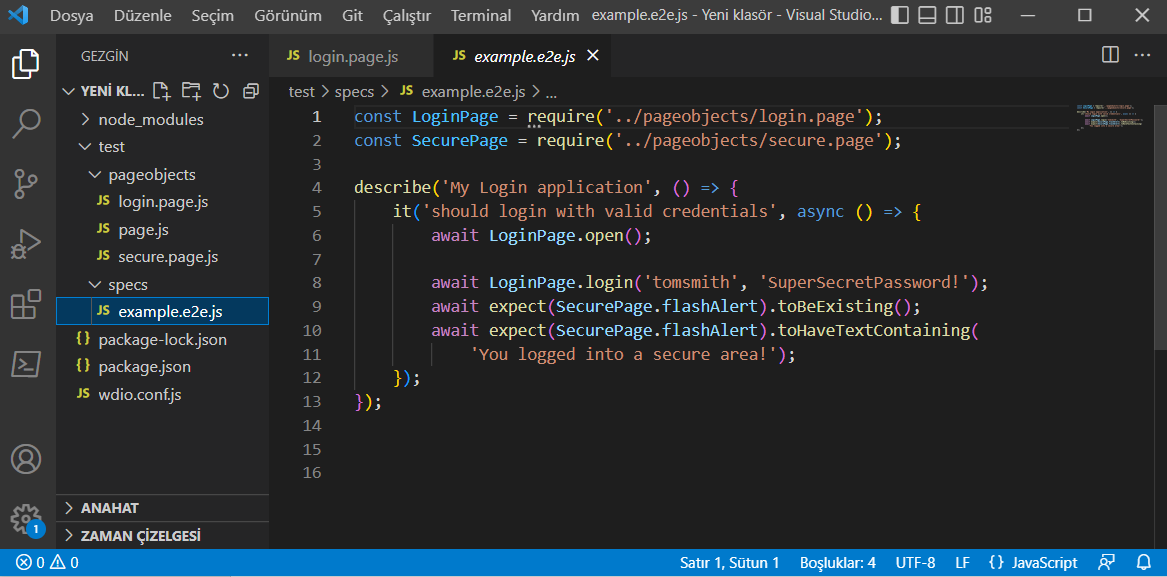
*SetValue()* methodu ile textboxın içi doldurulur. *(Bakınız Şekil 10)*

**Seçilen element checkbox ise;**

*Click();* ile alan seçilmiş olur.

Specs klasörü altında bulunan example.e2e.js dosyasına gidilir.

Üst kısımda pagelerin import edildiği görülür.



*Şekil 11 Proje sayfası*

Describe kısmında teste ait bilgiler eklenir ki bu sayede hangi test adımında hata var hangi test adımı geçti takip edilebilir.

İt kod bloğu içerisinde de yine alt adımlara ait tanımlamalar , bilgilendirme satırları eklenebilir. Fonksiyonlar it kod bloğu içerisinde çağırılır, parametre girdileri yapılabilir.

## Allure Reporter Kullanım Amacı?

Testler WebDriver.io ile yazılıp koşulduktan sonra raporlanmalıdır. Bu noktada Allure Framework devreye girer. Allure Framework, test adımları, parametreler, ekran görüntüleri ve logların eklenebildiği, çoklu dil desteği olan açık kaynak kodlu bir raporlama aracıdır. Allure ile yazılan testlerin sonuçları anlaşılır bir şekilde raporlanabilir.

5. Allure Reporter Kurulumu?

WebDriver.io kurulumu yapılan klasör terminal ekranında açılır. *(Bakınız Şekil 1)*

Aşağıdaki kodlar sırasıyla terminal ekranında çalıştırılır.

npm install @wdio/allure-reporter --save-dev

Metin Kutusu 1 Allure Kurulum

Klasör içerisindeki wdio.config.js dosyası açılır ve içerisine aşağıdaki kod eklenir.

exports.config = {

// ... reporters: [['allure', {

outputDir: 'allure-results',

disableWebdriverStepsReporting: true,

disableWebdriverScreenshotsReporting: true,

}]],

// ....

}

Metin Kutusu 2

Example.e2e.js dosyasının içinde üst kısıma aşağıdaki kod eklenir.

const allureReporter = require('@wdio/allure-reporter').default

*Metin Kutusu 3*

Example.e2e.js dosyası içinde it kod bloğu içerisine aşağıdaki kod eklenir.

allureReporter.addFeature('Feature');

*Metin Kutusu 4*

it('should login with valid credentials',async() => {

awaitLoginPage.open();

awaitLoginPage.login('tomsmith', 'SuperSecretPassword!'); awaitexpect(SecurePage.flashAlert).toBeExisting(); awaitexpect(SecurePage.flashAlert).toHaveTextContaining(

'You logged into a secure area!');

allureReporter.addFeature('Feature');

});

*Metin Kutusu 5*

Wdio.config.js dosyası açılır ve aşağıdaki kod en üst kod satırına eklenir.

const allure = require (‘allure-commandline’)

*Metin Kutusu 6*

Terminal ekranında aşağıdaki kod çalıştırılır.

npm i allure-commandline

*Metin Kutusu 8*

*Metin Kutusu 7*

Wdio.config.js dosyası içerisinde yorum satırları arasında onComplete fonksiyonu bulunur, yorum satırında çıkartılır ve aşağıdaki kod içerisine eklenir.

onComplete: function() {

const reportError = new Error('Could not generate Allure report')

const generation = allure(['generate', 'allure-results', '--clean'])

return new Promise((resolve, reject) => {

const generationTimeout = setTimeout(

() => reject(reportError),

5000)

generation.on('exit', function(exitCode) {

clearTimeout(generationTimeout)

if (exitCode !== 0) {

return reject(reportError)

}

console.log('Allure report successfully generated')

resolve()

})

})

},

*Metin Kutusu 8*

Allure Reporter kurulumu ve WebDriver.io ile bağlantısı tamamlandı. Allure result dosyasının otomatik oluşması için vs code da terminal ekranı açılır ve aşağıdaki kodla proje çalıştırılır.

npx wdio

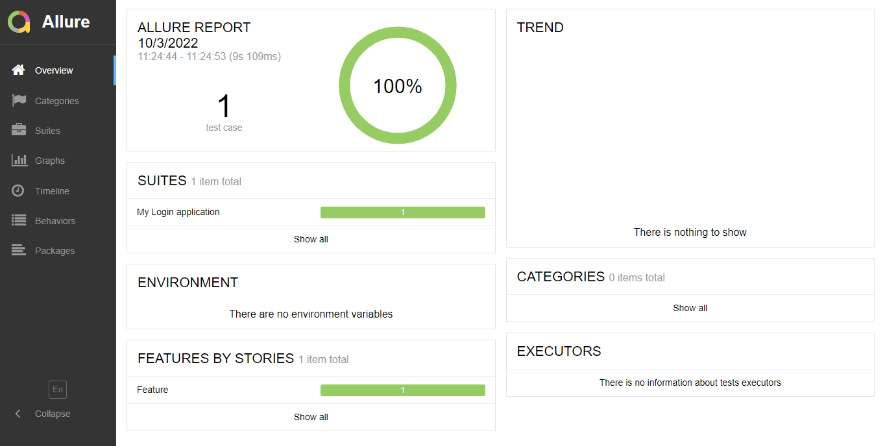
*Metin Kutusu 9*

Aşağıdaki kod terminal ekranında çalıştırılarak Allure Reporter başlatılır.

allure open

*Metin Kutusu 10*

Example.e2e.js dosyasının test koşum sonucu aşağıdaki gibi Allure Reporter ekranında görüntülenir.



*Şekil 12*