

**T.C.**

**FIRAT ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ**

**YAZILIM MÜHENDİSLİĞİ**

**YMU412 YAZILIM KALİTE GÜVENCESİ**

**Ödev Dokümantasyonu**

**Ödev Çalışma Grubu:**

180290608 - Mohammad Hajjar

180290603 - Nadem Ghnnam

180290611 - Ömer Akrum

**Ders Sorumlusu:**

DR. ÖĞR. ÜYESİ ÖZGÜR KARADUMAN

**İÇİNDEKİLER**

1. **SELENİUM** 
   1. Selenium genel anlatımı
   2. Selenium IDE anlatımı
   3. Selenium WebDriver anlatımı
   4. Selenium IDE testi
   5. Selenium WebDrive testi
   6. Selenium Grid anlatımı
2. **POSTMAN** 
   1. Interface nedir?
   2. API nedir?
   3. HTTP nedir?
   4. API testi nedir ve neden yapılmalıdır?
   5. Postman nedir?
   6. Neden postman?
   7. Proje notları
3. **APPIUM**
   1. Appium Nedir?
   2. Appium'un Özellikleri
   3. Appium'un Avantajları
   4. Appium'un Dezavantajları
   5. Appium’u kullanmak için ön koşulları
   6. Appium Nasıl Çalışır?
   7. Test Senaryosu ve Test Kodu

**4. KAYNAKÇA**

**1. SELENİUM**

**1.1. Selenium genel anlatımı**

**Selenium Nedir:**

Selenium tarayıcı üzerinden otomasyon yazmamıza yardımcı olan bir araçtır. Örneğin bir websiteniz var burada birden fazla butonlar ve fonksiyonlar bulunmakta. Siz bu sitedeki bütün fonksiyonları kendiniz test etmek yerine Selenium ile bir tane otomasyon yapıp webdriver, Selenium IDE ve Selenium Grid kullanarak tek tek işlemleri sanal tarayıcı ile izleyip bu fonksiyonların sıkıntı yaratıp yaratmadığını görebilirsiniz [1].

**Desteklenenler:**

* Desteklenen İşletim Sistemleri : Windows, MacOS, Linux, Unix.
* Desteklenen Tarayıcılar : IE, Firefox, Chrome, Opera, Safari.
* Desteklenen Programlama Dilleri : Java, Python, C#, PHP, Ruby, Perl, JavaScript [2].

**Avantajları:**

* **Web'e hazır:**

Güvenilir uçtan uca testleri hızlı bir şekilde geçirmek için basit bir şekilde. Hiç kod yazmaya ihtiyaç olmadan veya basit kodlar yazarak halledebiliriz.

* **Kolay hata ayıklama**:

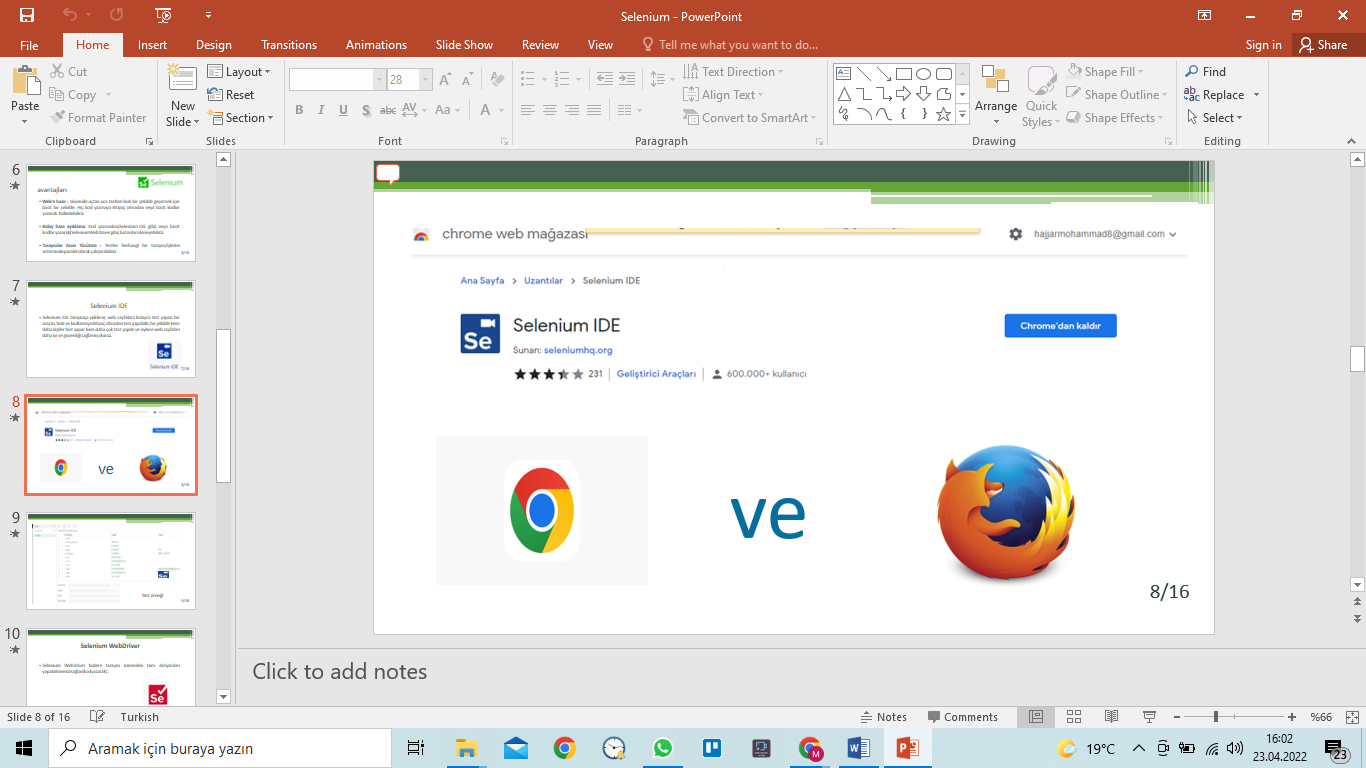
Kod yazmadan(Selenium IDE gibi) veya basit kodlar yazarak(Selenium WebDriver gibi) botonları deneyebiliriz.

* **Tarayıcılar Arası Yürütme:**

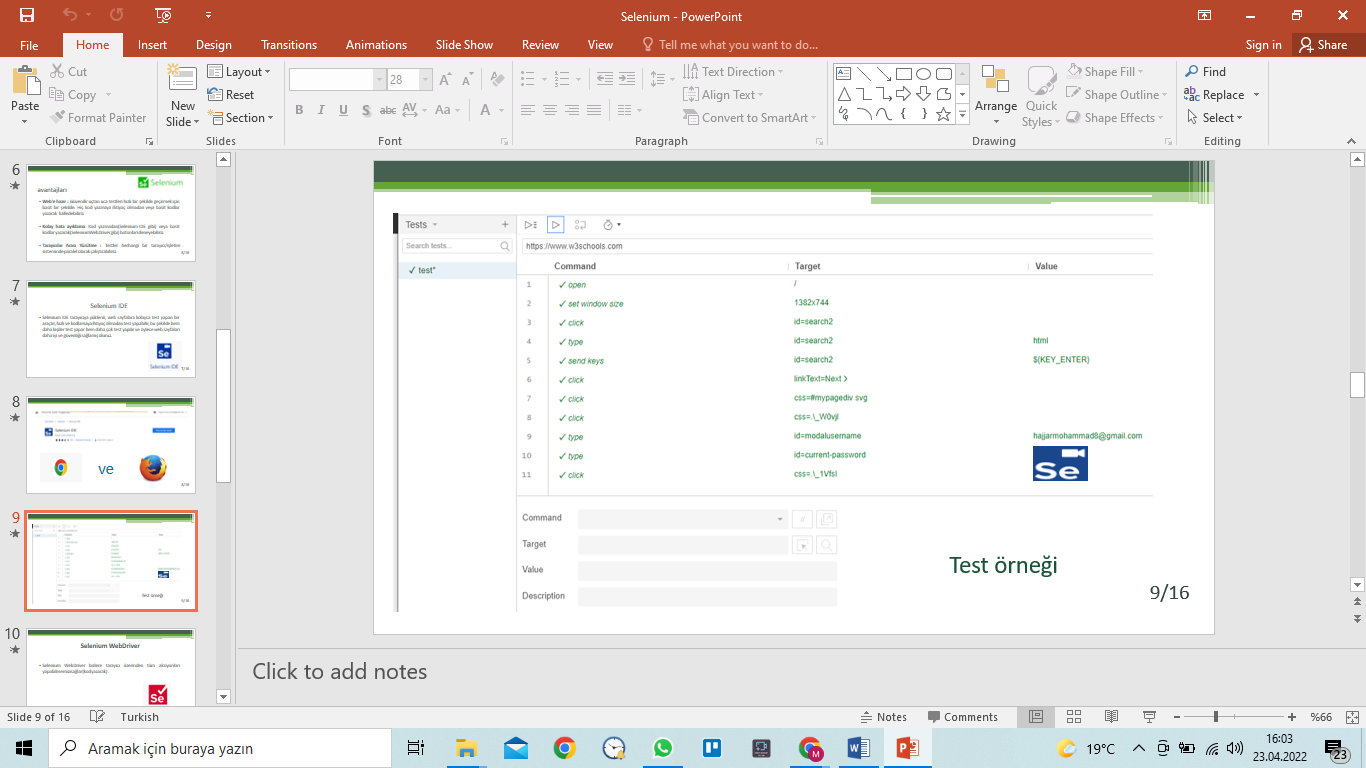
Testlerherhangi bir tarayıcı/işletim sisteminde paralel olarak çalıştırabiliriz [3].

**1.2. Selenium IDE anlatımı**

Selenium IDE tarayıcıya yüklenir, web sayfalara kolayca test yapan bir araçtır, hızlı ve kodlamaya ihtiyaç olmadan test yapabilir, bu şekilde hem daha kişiler test yapar hem daha çok test yapılır ve öylece web sayfaları daha iyi ve güvenliği sağlamış oluruz [5].



Şekil1. Selenium IDE tarayıcıya yüklenen adres [4].



Şekil2. Selenium IDE test örneği.

**1.3. Selenium WebDriver anlatımı**

Selenium WebDriver bizlere tarayıcı üzerinden tüm aksiyonları yapabilmemizi sağlar(kod yazarak) [6].

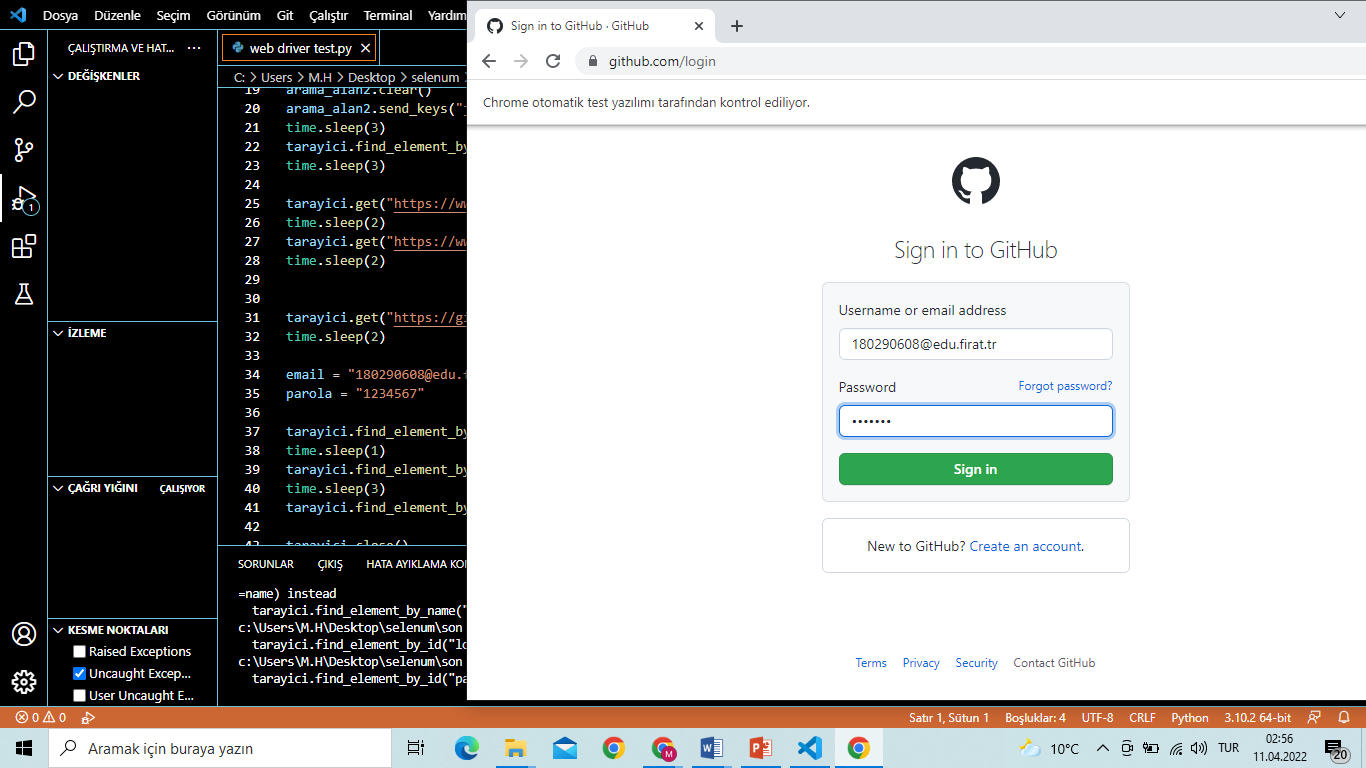
**Kulanılabilir programalma diller:**

* Java
* Ruby
* C#
* Python
* JavaScript

**Kulanılabilir tarayıcılar:**

* Firefox
* Chrome
* Safari
* Opera
* Edge

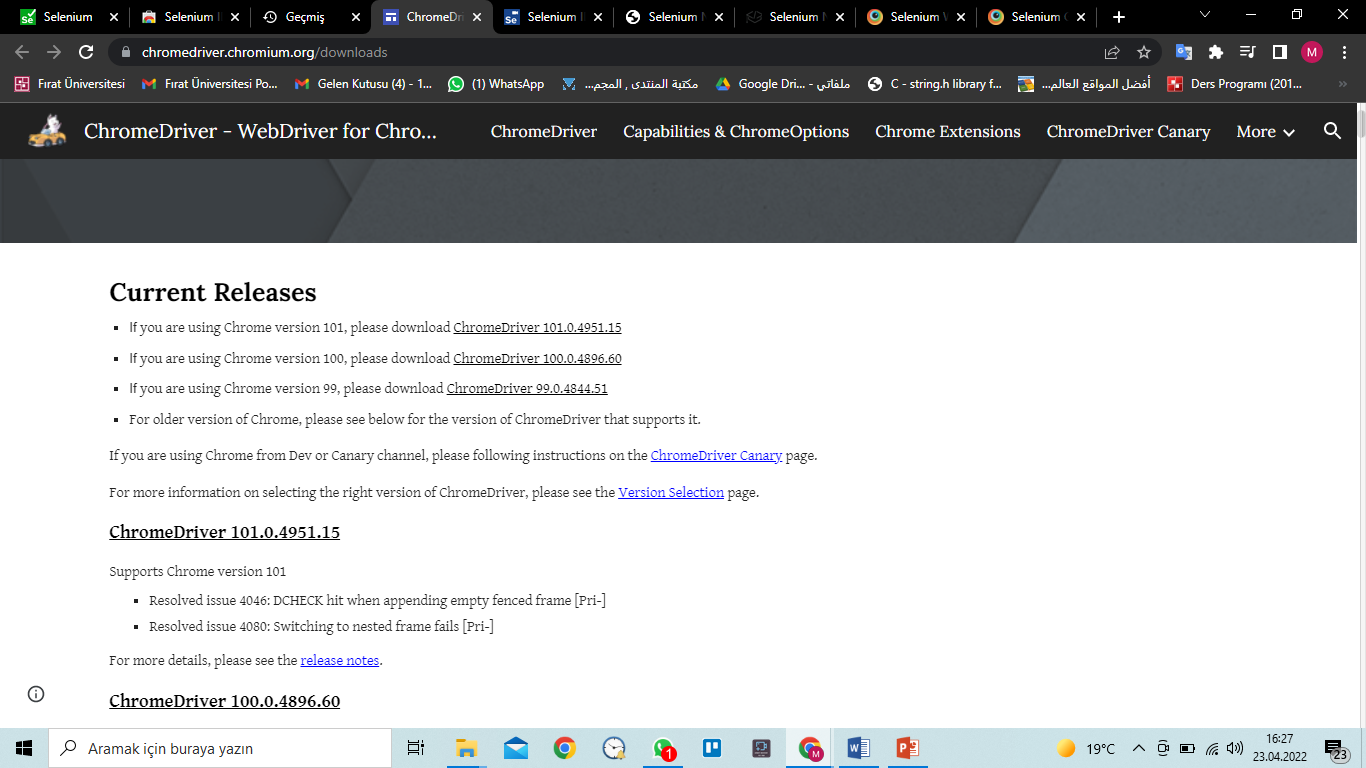
**1.4. Selenium IDE testi**



Şekil3. Selenium WebDriver test örneği.

**Test çalışması:**

ChromeDriver indermesi gerekir:



Şekil4. Selenium WebDriver, Chrome arasında iletişimi sağlamamız için Chrome versiyonu [7].

**1.5. Selenium WebDrive testi**

**Test çalışması:**

Kod satırlarını yanına yanıtlar olarak açıklamalarını yazdım. Test çalışması(ekran çıktısı) hakkında konuşacaksak ;

1. Google adresine geçer.
2. Ordaki boş alanı içinde “Hello world” yazar.
3. Sonra yazdığımız değeri dödürür.
4. Boş alandaki “Hello world” siler yerine “java” yazar.
5. Yazdığımız değeriye entera basar.
6. Görsellere geçer sonra bir foto seçer.
7. Github Login sayfasına geçer orda verdiğimiz değerleri dener boş alanlarda en sonda botuna basar.

**Webdriver kod testi:**

from selenium import webdriver

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.common.keys import Keys

import time

tarayici = webdriver.Chrome('./chromedriver') #nesne oluşturarak chrome ile iletişimi kurduk.

tarayici.get("https://www.google.com") # google adresine gidiyoruz

print(tarayici.title) #ekrana çıkartıyoruz

arama\_alan1 = tarayici.find\_element\_by\_name("q") #arama boş alanı adıyla kullandık

arama\_alan1.clear() #boş alanı yazı varsa temizle dedik

arama\_alan1.send\_keys("hello world") # hello world yaz

time.sleep(3) #3 saniye dur sonra devam et

arama\_alan1.send\_keys(Keys.RETURN) #boş alana verdiğimiz değeri döndür

time.sleep(3)

arama\_alan2 = tarayici.find\_element\_by\_name("q") #yine den çağır

arama\_alan2.clear() #önceki yazı temizle

arama\_alan2.send\_keys("java") #boş alanda java yaz

time.sleep(3)

tarayici.find\_element\_by\_name("q").send\_keys(Keys.ENTER) # enter e bas

time.sleep(3)

tarayici.get("https://www.google.com/search?q=java&rlz=1C1GCEA\_enTR994TR994&tbm=isch&sxsrf=APq-WBtN4fwhHDCMZQNQ8hyWyzm1s2FJOg:1649493358119&source=lnms&sa=X&ved=2ahUKEwjV3q3NyYb3AhU2REEAHSlaCO4Q\_AUoAXoECAEQAw&biw=944&bih=625&dpr=1")

time.sleep(2)

tarayici.get("https://www.google.com/search?q=java&rlz=1C1GCEA\_enTR994TR994&tbm=isch&sxsrf=APq-WBtN4fwhHDCMZQNQ8hyWyzm1s2FJOg:1649493358119&source=lnms&sa=X&ved=2ahUKEwjV3q3NyYb3AhU2REEAHSlaCO4Q\_AUoAXoECAEQAw&biw=944&bih=625&dpr=1#imgrc=oQsIfeD6p6\_vIM")

time.sleep(2)

tarayici.get("https://github.com/login")

time.sleep(2)

email = "180290608@edu.firat.tr"

parola = "1234567"

tarayici.find\_element\_by\_id("login\_field").send\_keys(email)# id yi kullanarak boş alana mail değeri aktarıyoruz.

time.sleep(1)

tarayici.find\_element\_by\_id("password").send\_keys(parola) # id yi kullanarak boş alana parola değeri aktarıyoruz.

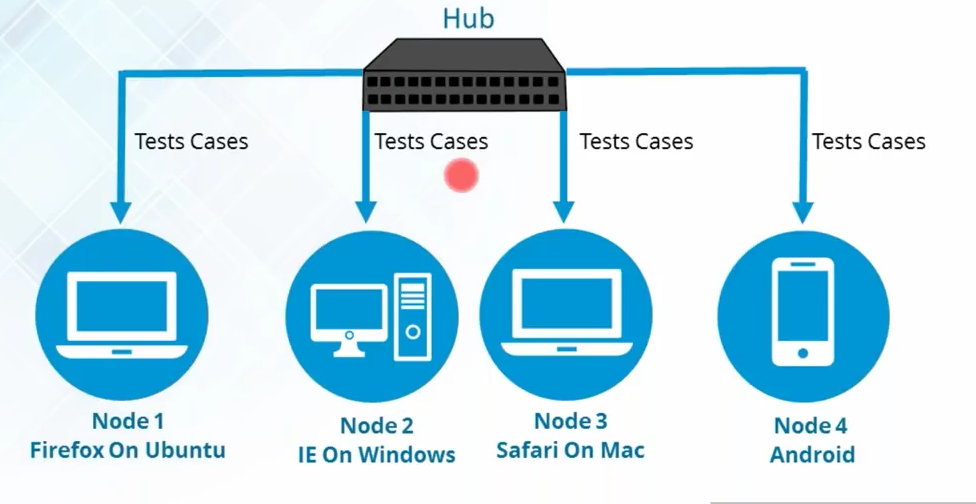
time.sleep(3)

tarayici.find\_element\_by\_name("commit").click() #bottuna bas diyoruz.

tarayici.close() #pencereyi (nesneyi) kapat.

**1.6. Selenium Grid anlatımı**

Selenium Grid oluşturulan test caselerinin farklı ortamlarda aynı anda koşulmasını sağlamaktadır. Farklı tarayıcılarda farklı işletim sistemlerda gerçekleştirilebilir [8].



Şekil5. selenium grid [9].

**2.POSTMAN**

**2.1. Interface nedir?**

İnsanlar ve makineler arasındaki etkileşimlerin gerçekleştiği alandır. Bu etkileşimin amacı, makinenin eş zamanlı olarak operatörlere karar verme sürecine yardımcı olan bilgileri geri beslerken, makinenin insan tarafından etkin bir şekilde çalıştırılmasına ve kontrolüne izin vermektir.

**2.2. API nedir?**

API, Türkçede *“Uygulama Programlama Arayüzü”* anlamına gelen *“Application Programming Interface”* tanımının baş harflerinden ortaya çıkmış bir terimdir. Kısa bir tanımla API, kendine ait veriler ve çalışma prensipleri ile geliştirilmiş uygulamaların, birbirileri ile iletişime geçerek çalışmasını mümkün kılan yazılımdır.

**2.3. HTTP nedir?**

*HTTP(Hypertext Transfare Protocol):*

WWW, istemcileri (clint)ve sunucular(server) arasındaki iletişimle ilgilidir.

İstemci bilgisayarlar(clint) ve web sunucuları (servers) arasındaki iletişim,

HTTP İstekleri gönderilerek ve HTTP Yanıtları alınarak yapılır.

İstemciler (clients) genellikle tarayıcılar dır (Chrome, Edge, Safari), ancak herhangi bir program veya cihaz türü olabilirler.

Sunucular(servers) genellikle buluttaki bilgisayarlardır**.**

*HTTP durum kodlarının listesi:*

* 201 — Başarılı istek ve veriler için.
* 204 — Boş cevaplar için.
* 400 — Kötü cevap için. İstek anlaşılamadı veya gerekli parametreler eksikti.
* 401 — Yetkisiz erişim için. Kimlik doğrulama başarısız oldu veya kullanıcının istenen işlem için izinleri yok.
* 403 — Yasak için. Erişim reddedildi.
* 404 — Bilgi bulunamadığı için.
* 405 — İstek desteklenmediği veya metoda izin verilmediği için.
* 500 — Dahili Sunucu Hatası için.
* 503 — Kullanılamayan Servis İçin.

**2.4. API testi nedir ve neden yapılmalıdır?**

API testi, API’lerin servislerle olan entegrasyonunun test edilmesidir. Geliştirmede büyük bir soruna neden olacak bir hatayı gözden kaçırırsak bu hatayı üretim ortamında ayıklamak daha zor olacaktır.

**2.5. Postman nedir?**

Postman, uzun kodlara ihtiyaç duymadan API'larımızı kolayca test edebileceğimiz, API'mızın dokümasyonunu hazırlamamızı sağlayan ve "rest client" olarak adlandırabileceğimiz bir araçtır. Postman ayrıca Chrome için bir eklenti olarak da kullanılabilir ve disktop uygulama olarak da kullanabilir.

**2.6. Neden postman?**

* Postacı, aşağıdakiler gibi birçok gelişmiş özellik sahiptir:
* API testi.
* API geliştirme.
* Hala geliştirilmekte olan API'ler için Mock uç noktaları ayarlama.
* API belgeleri.
* API uç noktası yürütmesinden alınan yanıtlara ilişkin iddialar.
* Jenkins, TeamCity gibi CI-CD araçlarıyla entegrasyon.
* API testlerinin yürütülmesini otomatikleştirme vb.

**2.7. Proje notları**

**Simple Books API**

This API allows you to reserve a book.

The API is available at https://simple-books-api.glitch.me

**Endpoints**

**Status**

GET /status

Returns the status of the API.

**List of books**

GET /books

Returns a list of books.

Optional query parameters:

* type: fiction or non-fiction
* limit: a number between 1 and 20.

**Get a single book**

GET /books/:bookId

Retrieve detailed information about a book.

**Submit an order**

POST /orders

Allows you to submit a new order. Requires authentication.

The request body needs to be in JSON format and include the following properties:

* bookId - Integer - Required
* customerName - String - Required

Example

POST /orders/

Authorization: Bearer <YOUR TOKEN>

{

"bookId": 1,

"customerName": "John"

}

The response body will contain the order Id.

**Get all orders**

GET /orders

Allows you to view all orders. Requires authentication.

**Get an order**

GET /orders/:orderId

Allows you to view an existing order. Requires authentication.

**Update an order**

PATCH /orders/:orderId

Update an existing order. Requires authentication.

The request body needs to be in JSON format and allows you to update the following properties:

* customerName - String

Example

PATCH /orders/PF6MflPDcuhWobZcgmJy5

Authorization: Bearer <YOUR TOKEN>

{

"customerName": "John"

}

**Delete an order**

DELETE /orders/:orderId

Delete an existing order. Requires authentication.

The request body needs to be empty.

Example

DELETE /orders/PF6MflPDcuhWobZcgmJy5

Authorization: Bearer <YOUR TOKEN>

**API Authentication**

To submit or view an order, we need to register our API client.

POST /api-clients/

The request body needs to be in JSON format and include the following properties:

* clientName - String
* clientEmail - String

Example

{

"clientName": "Postman",

"clientEmail": "valentin@example.com"

}

The response body will contain the access token. The access token is valid for 7 days.

**Possible errors**

Status code 409 - "API client already registered." Try changing the values for clientEmail and clientName to something else.

**3. APPIUM**

**3.1. Appium Nedir?**

Appium , iOS mobil, Android mobil ve Windows masaüstü platformlarında yerel, mobil web ve hibrit uygulamalarında web sürücüsü kullanarak test otomasyonunda kullanılan açık kaynak kodlu bir test aracıdır [12].

Günümüzün gelişim alanında mobil uygulamalar web uygulamalarına göre daha fazla kullanılmaktadır. Bu nedenle, mobil yazılım otomasyon test teknolojisi hakkında bilgi sahibi olmak ve yeni teknoloji ile bağlantıda kalmak çok önemlidir [13].

**3.2. Appium'un Özellikleri**

* Appium, uygulama kaynak kodu veya kitaplık gerektirmez.
* Appium, güçlü ve aktif bir topluluk sağlar.
* Appium'un çoklu platform desteği vardır, yani aynı test senaryolarını birden fazla platformda çalıştırabilir.
* Appium, test komut dosyalarının paralel yürütülmesine izin verir.
* Appium'da küçük bir değişiklik, uygulamanın yeniden yüklenmesini gerektirmez.
* Appium, Java, Python, C#, Ruby, PHP, node.js ile JavaScript ve Selenium istemci kitaplığına sahip diğer birçok dil gibi çeşitli dilleri destekler.

**3.3. Appium'un Avantajları?**

* Appium açık kaynaklı bir araçtır, yani ücretsiz olarak kullanılabilir. Yüklemek kolaydır.
* Hibrit, yerel ve web uygulamalarının otomatik olarak test edilmesini sağlar.
* Diğer test araçlarının aksine, Appium'u otomasyonla uyumlu hale getirmek için uygulamanıza herhangi bir ek aracı eklemeniz gerekmez. App Store'a yüklenecek olan aynı uygulamayı test eder.
* Appium, platformlar arası mobil test yapmamıza olanak tanıyan, platformlar arası, ücretsiz olarak kullanılabilen bir mobil test aracıdır. Bu, birden fazla platformda test edebileceğiniz anlamına gelir (hem Android hem de IOS platformları için tek API).

**3.4. Appium'un Dezavantajları?**

* Ayrıntılı raporların eksikliği.
* Testler uzak web sürücüsüne bağlı olduğundan, biraz yavaştır.

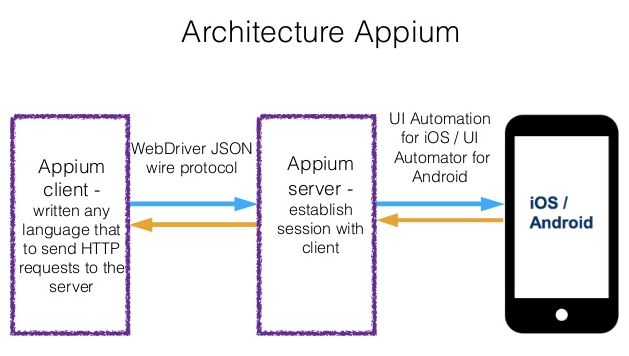
**3.5. Appium’u kullanmak için ön koşulları?**

* Java (JDK)
* Android Studio
* Ek Android SDK araçlarını
* Appium jar dosyasını
* Appium Masaüstü İstemcisini
* Eclipse IDE

**3.6. Appium Nasıl Çalışır?**

Appium, kurulumu bir REST API’sini temsil eden ve Selenium WebDriver’ı uygulayan Node.js platformunda yazılmış bir HTTP sunucusudur.Bir istemci sunucu mimarisi üzerinde çalışır. Appium, testleri çalıştırmak için mevcut WebDriver istemcisini kullanmanıza izin verir [14].

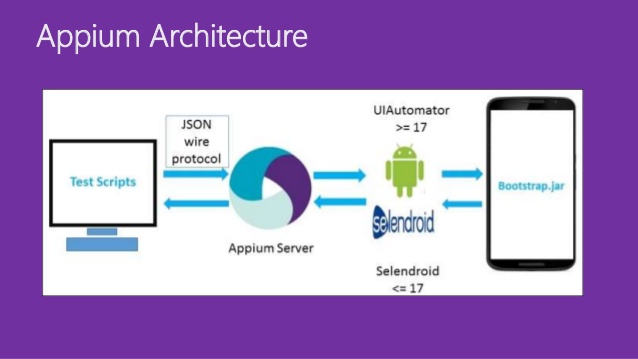
Appium sunucusu, Appium istemcisinden gelen talepleri işler ve bunları, test komut dosyalarının otomatikleştirildiği simülatör / emülatör / gerçek cihaza iletir. Testin sonuçları, bir sunucu yanıtı şeklinde Appium sunucusu aracılığıyla Appium istemcisine iletilir[14].



Şekil 6.Appium'un temel mimarisi [15].

Appium, Android cihazlardaki uygulamaları otomatikleştirmek için Android kullanıcı arayüzünü test etmek için kullanılan UIAutomator çerçevesini (veya Selendroid) kullanır.

* Android API 17 veya üstü için kullanılan UIAutomator.
* Android API 17 veya daha azı için kullanılan Selendroid.



Şekil 7.Appium sunucusu ve Android cihazlar çerçevesi [16].

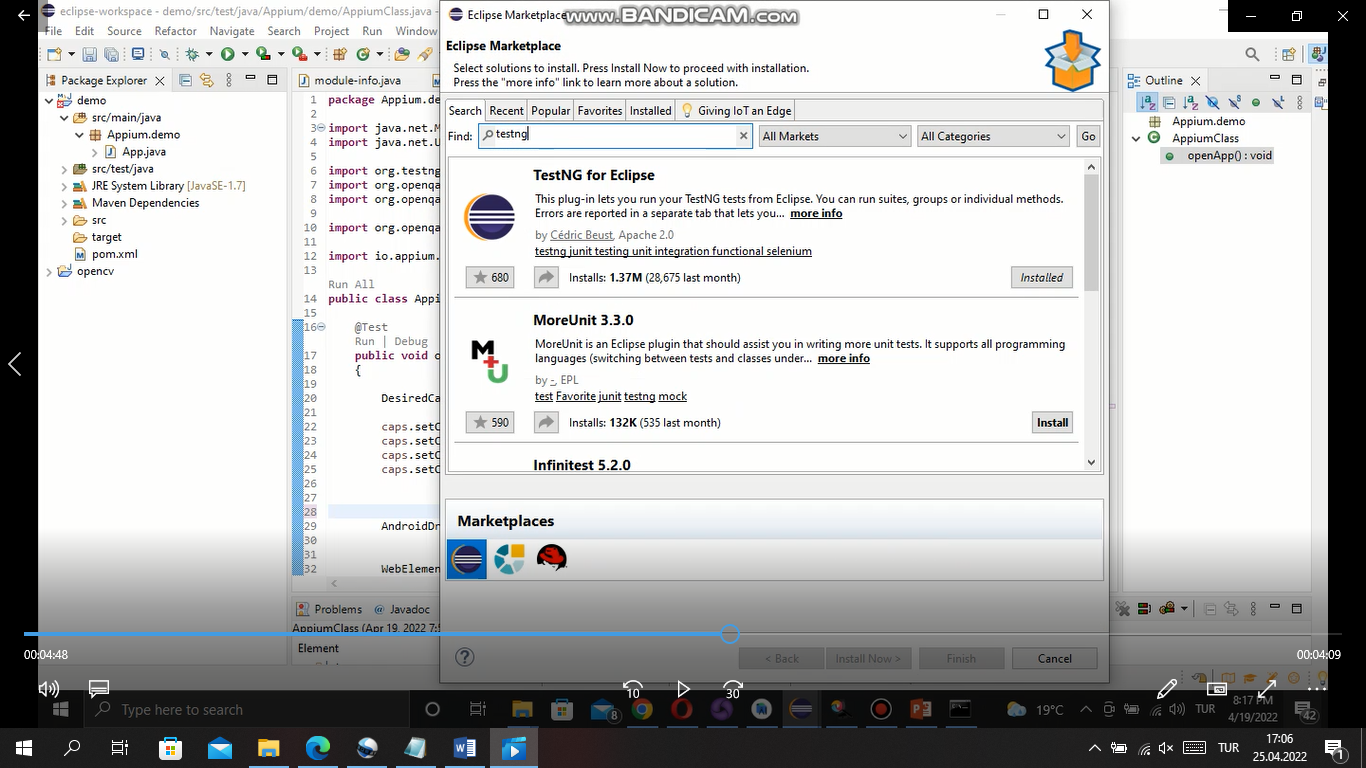
**3.7. Test Senaryosu ve Test Kodu**

komut istemine Emülatörü aktif hale geldi durumda olduğunu kontrol etmek için adb devices yazmamız gerekiyordur.

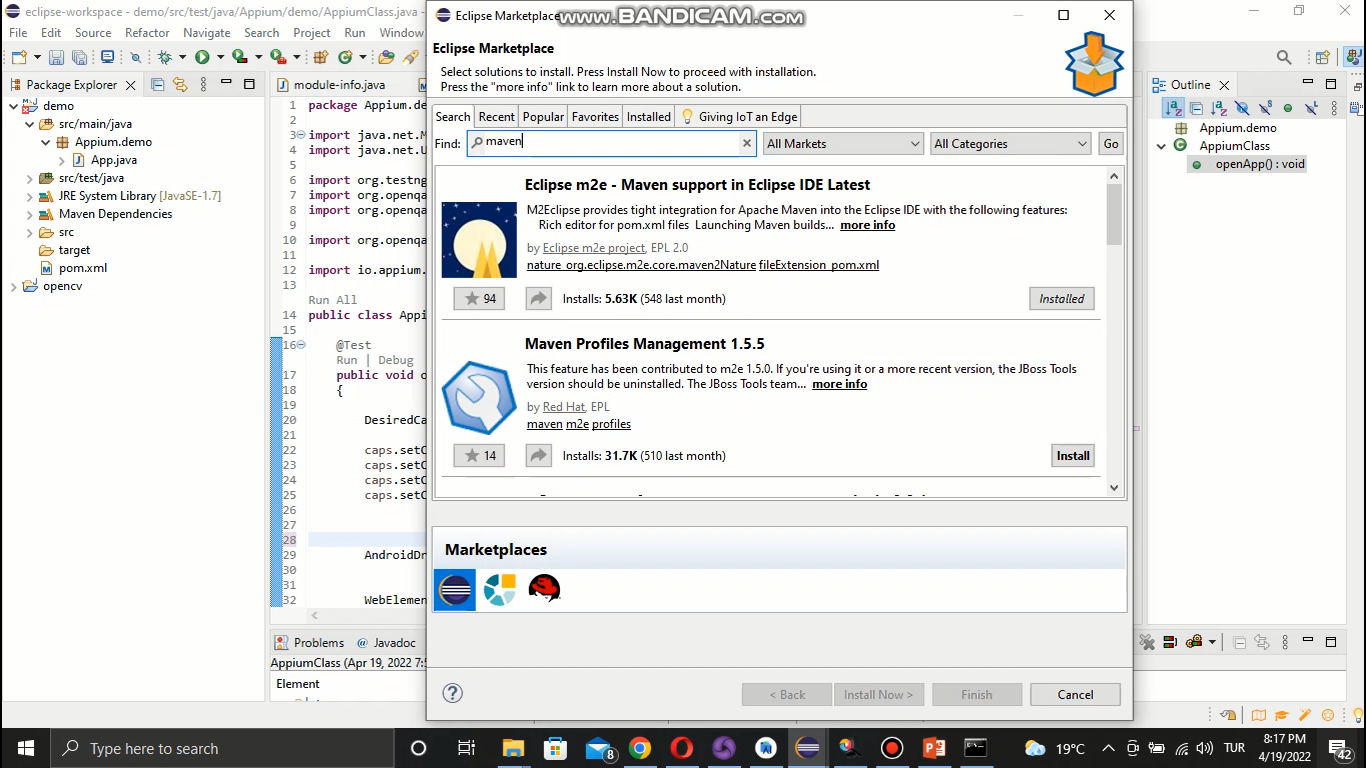


Şekil 8.komut istemi

TestNG ve maven eklentisileri Eclips'te Şekil 9 ve 10'deki gibi indirmemiz gerekiyordur.

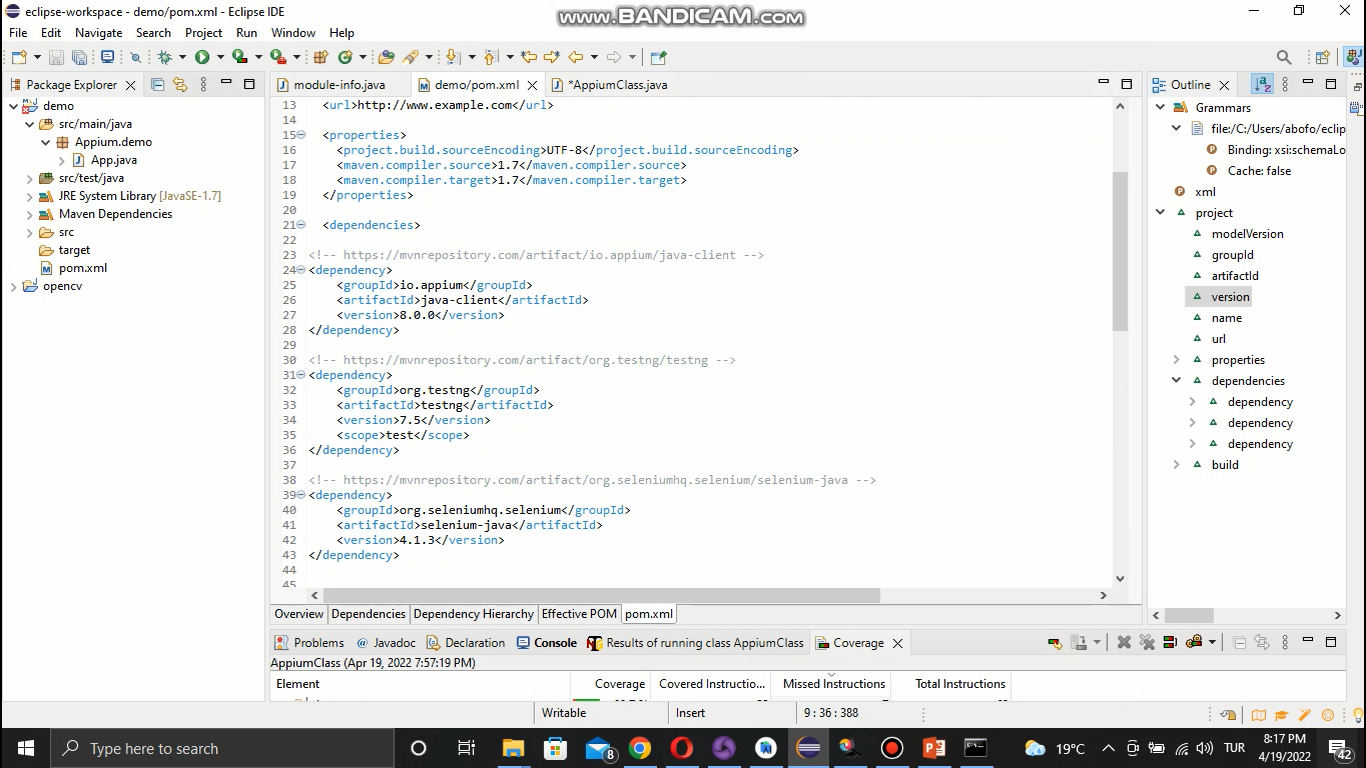


Şekil 9. Eclipse MarketPlace



Şekil 10. Eclipse MarketPlace

XML dosyasında, java-client ,testng ve selenium-java dependency vardır.



Şekil 11. pom.xml dosyası

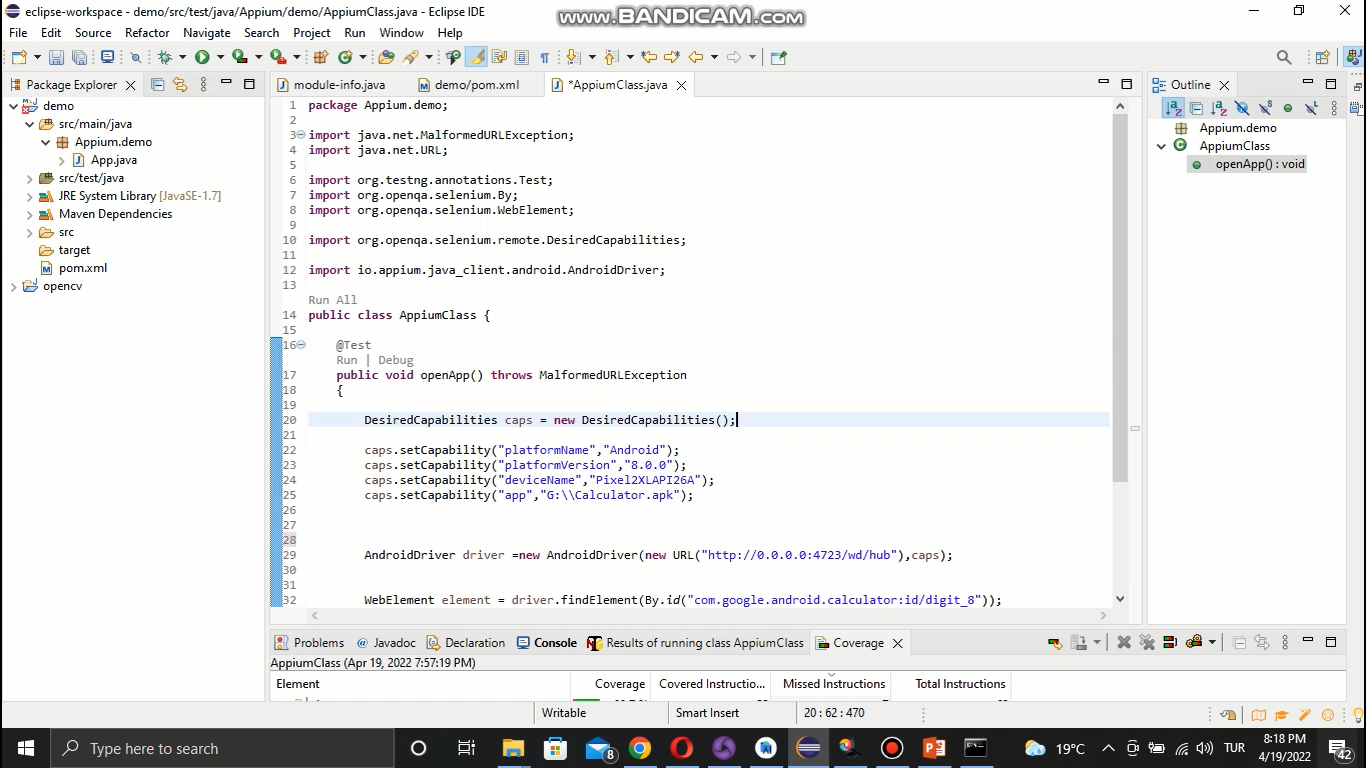
Şekil 12’de AppiumClass adında bir sınıf vardır. @Test Bu, projenin TestNG'sini çalıştırmak için TestNG'nin Automationdır. Kodu çalıştırmak için openApp bir fonksiyonu vardır. Desired Capabilitiesden caps adında bir nesne olusturmamız ve bir JSON nesnesinde kodlanmış anahtarlar ve değerleri tanımlamamız gerekiyordur.

• platformName: Hangi mobil işletim sistemi platformunu kullanacaktır.

• platformVersion: Mobil işletim sistemi sürümüdür.

• deviceName: Kullanılacak mobil cihaz veya emülatörün adıdır.

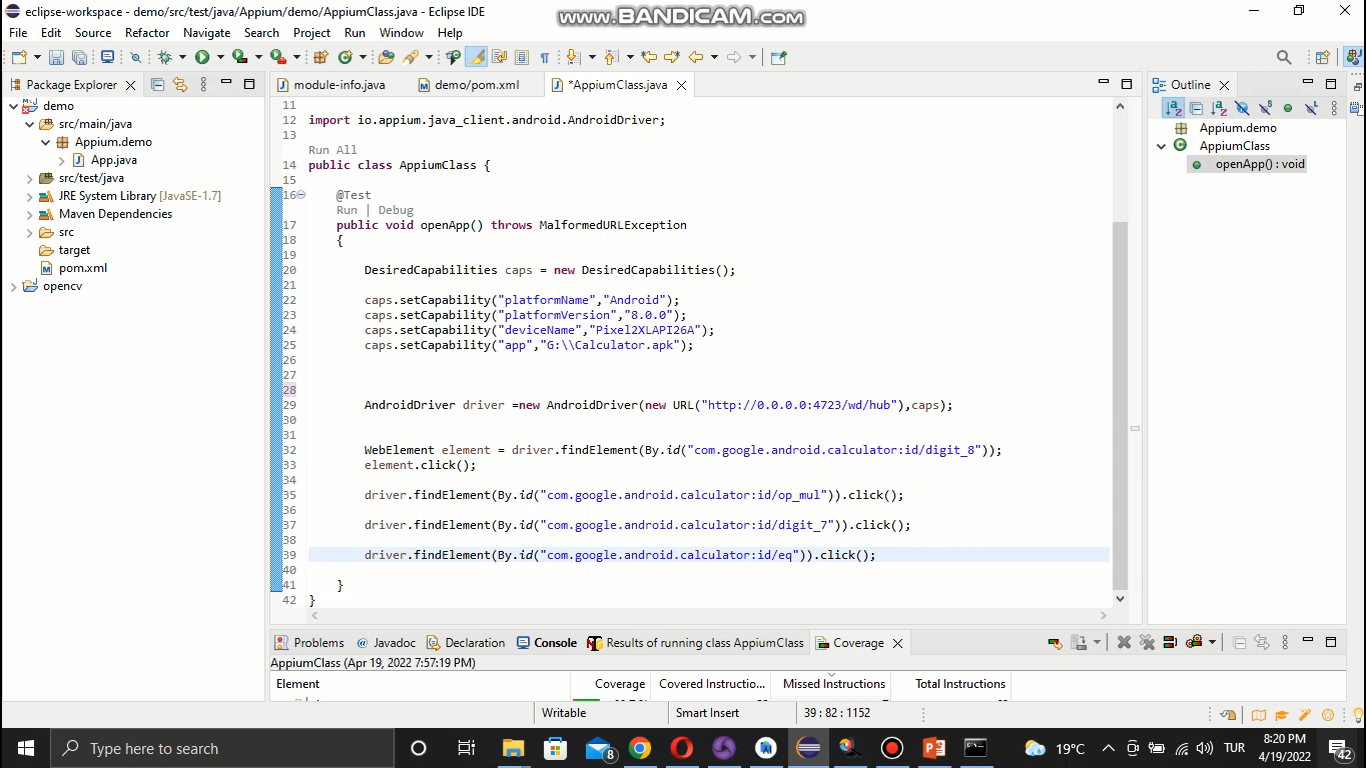
• app: Test edilecek uygulamadır.



Şekil 12. AppiumClass.java dosyası

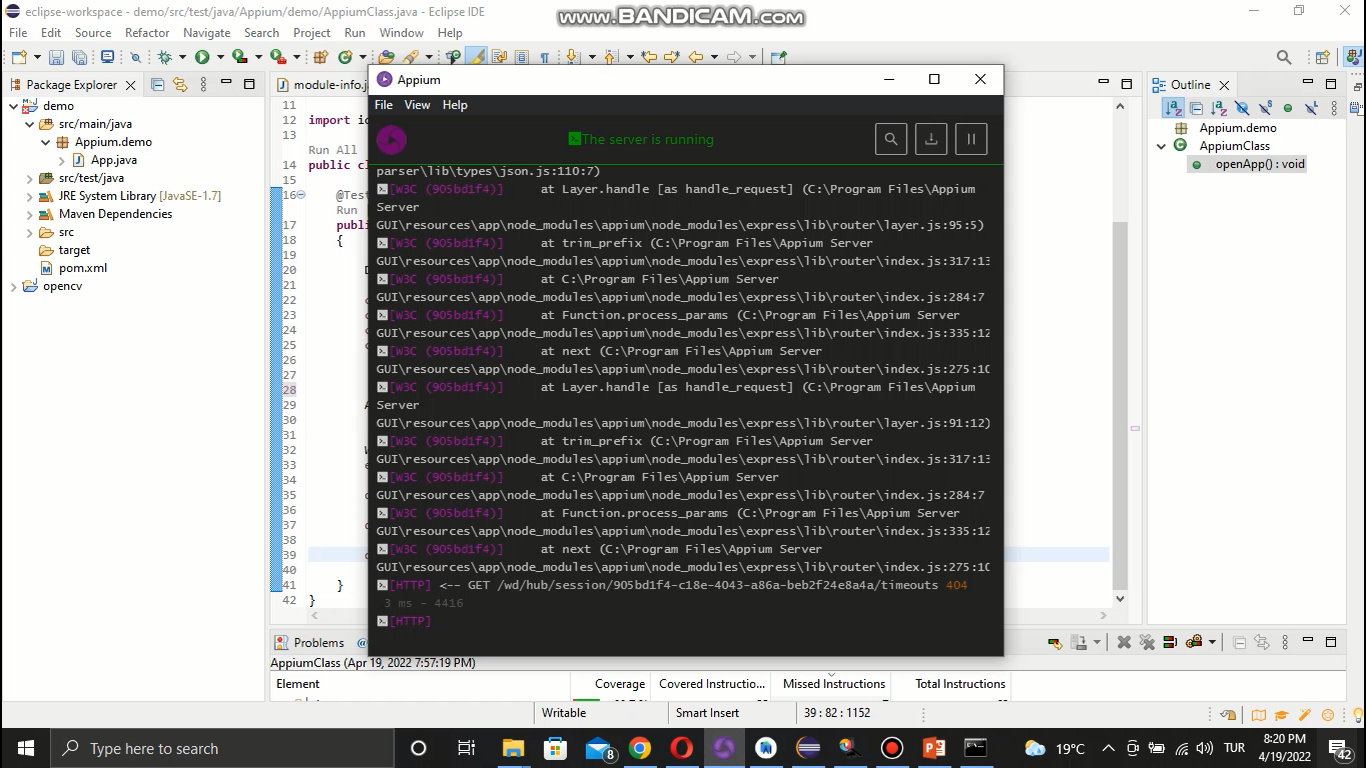
Programı appium sunucusu ve capabilite ile bağlamak için AndroidDriverden driver adında bir nesne olusturmamız ve WebElementten element adında bir nesne olusturmamız ve bu elementi ile id adısı seçme findElement fonksiyonu kullanarak ve üzerine tıklamak click fonksiyonu kullanarak gerekiyordur.

Şekil 13'te 32. satırda 8 butonunun id adıdır. 35, 37 ve 39. satırlarda çarpı, 7 ve eşittir butonunun id adları vardır.



Şekil 13. AppiumClass.java dosyası

Şekil 14'deki gibi Appium çalıştırdıktan sonra kodu çalıştırıyoruz.



Şekil 14. Appium

**Test kodu:**

package Appium.demo;

import java.net.MalformedURLException;

import java.net.URL;

import org.testng.annotations.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.remote.DesiredCapabilities;

import io.appium.java\_client.android.AndroidDriver;

public class AppiumClass {

Test@

public void openApp() throws MalformedURLException

{

DesiredCapabilities caps = new DesiredCapabilities();

caps.setCapability("platformName","Android");

caps.setCapability("platformVersion","8.0.0");

caps.setCapability("deviceName","Pixel2XLAPI26A");

caps.setCapability("app","G:\\Calculator.apk");

AndroidDriver driver =new AndroidDriver(new URL("http://0.0.0.0:4723/wd/hub"),caps);

WebElement element = driver.findElement(By.id("com.google.android.calculator:id/digit\_8"));

element.click();

driver.findElement(By.id("com.google.android.calculator:id/op\_mul")).click();

driver.findElement(By.id("com.google.android.calculator:id/digit\_7")).click();

driver.findElement(By.id("com.google.android.calculator:id/eq")).click();

{

{

**KAYNAKÇA**

[1] <https://www.mobilhanem.com/selenium-nedir-ve-selenium-ile-test-otomasyonu/>

[2] <https://medium.com/@ilkebasalak/selenium-nedir-8c7d908c93e6>

[3] <https://www.selenium.dev/>

[4] <https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd?hl=tr>

[5] <https://www.selenium.dev/selenium-ide/>

[6] <https://www.mobilhanem.com/selenium-nedir-ve-selenium-ile-test-otomasyonu/>

[7] <https://chromedriver.chromium.org/downloads>

[8] <https://www.mobilhanem.com/selenium-nedir-ve-selenium-ile-test-otomasyonu/>

[9] <https://semihsaydam.medium.com/selenium-grid-8e33e4c1d198>

[10] <https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md>

[11] <https://en.wikipedia.org/wiki/Interface_(matter)#:~:text=From%20Wikipedia%2C%20the%20free%20encyclopedia,and%20studied%20in%20surface%20science>.

[12] [https://medium.com/neslinecipoglu/appium-nedir-yeni-ba%C5%9Flayanlar-i%C3%A7in-kurulum-macos-c207cfd0dfb2#:~:text=Appium%20Nedir?%20Appium%20;%20iOS%20mobil,%20Android%20mobil,JavaScript,%20Node.js,%20PHP,%20Python,%20Ruby,%20C#,%20Clojure,%20Perl](https://medium.com/neslinecipoglu/appium-nedir-yeni-ba%C5%9Flayanlar-i%C3%A7in-kurulum-macos-c207cfd0dfb2#,%20Clojure,%20Perl).

[13] <https://tr.linkedin.com/pulse/appium-nedir-ne-i%C5%9Fe-yarar-abdulsamed-bah%C3%A7aci>

[14] <https://berkaydinc42l.medium.com/appium-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-84a17e309f53>

[15] <https://image.slidesharecdn.com/appium-150407053935-conversion-gate01/95/appium-3-638.jpg?cb=1428396695>

[16] <https://www.inviul.com/wp-content/uploads/2019/07/Appium-Server-Architecture-Android.jpg>

[17] <https://www.youtube.com/watch?v=yTW7hZZfTYo&list=PLUDwpEzHYYLsx_2JFNBMITjHqTnuszhb_>

[18] <https://mesutbeysulen.medium.com/mobil-test-otomasyonunda-appium-2e08406da150>

[19] <https://www.javatpoint.com/appium#:~:text=Appium%20is%20an%20open-source%20automation%20mobile%20testing%20tool%2C,same%20test%20to%20be%20run%20on%20multiple%20platforms>.

[20] <https://www.swtestacademy.com/appium-mobil-otomasyon/>