



Selenium

Page Object Model Framework design

17 ARALIK 2020
DERS 20

TECHPROED

Mehmet BULUTLUOZ
Elektronik Muh.

Java'dan Hatırlamamız Gerekenler

Baska bir Class'dan variable veya method kullanmak istersek 3 yontem kullanabiliriz

A- inheritance (Miras) kullandigimiz Class'i extends anahtar kelimesi (key word) ile istedigimiz Class'in child'i yapabiliriz. Bu durumda object olusturmaya gerek kalmadan Parent class'a ulasabilir ve oradaki variable ve methodlari kullanabiliriz. (Test Base gibi)

B- Object olusturarak istedigimiz class'a ulasabilir ve o class'daki variable ve methodlari object'imizi araciligiyla kullanabiliriz

ornek: Okul class'ina ulasmak istiyorsak

Okul okul=new Okul(); diyerek object uretir, sonra

okul.variable diyerek variable'lara

okul.method() diyerek de methodlara ulasabiliriz

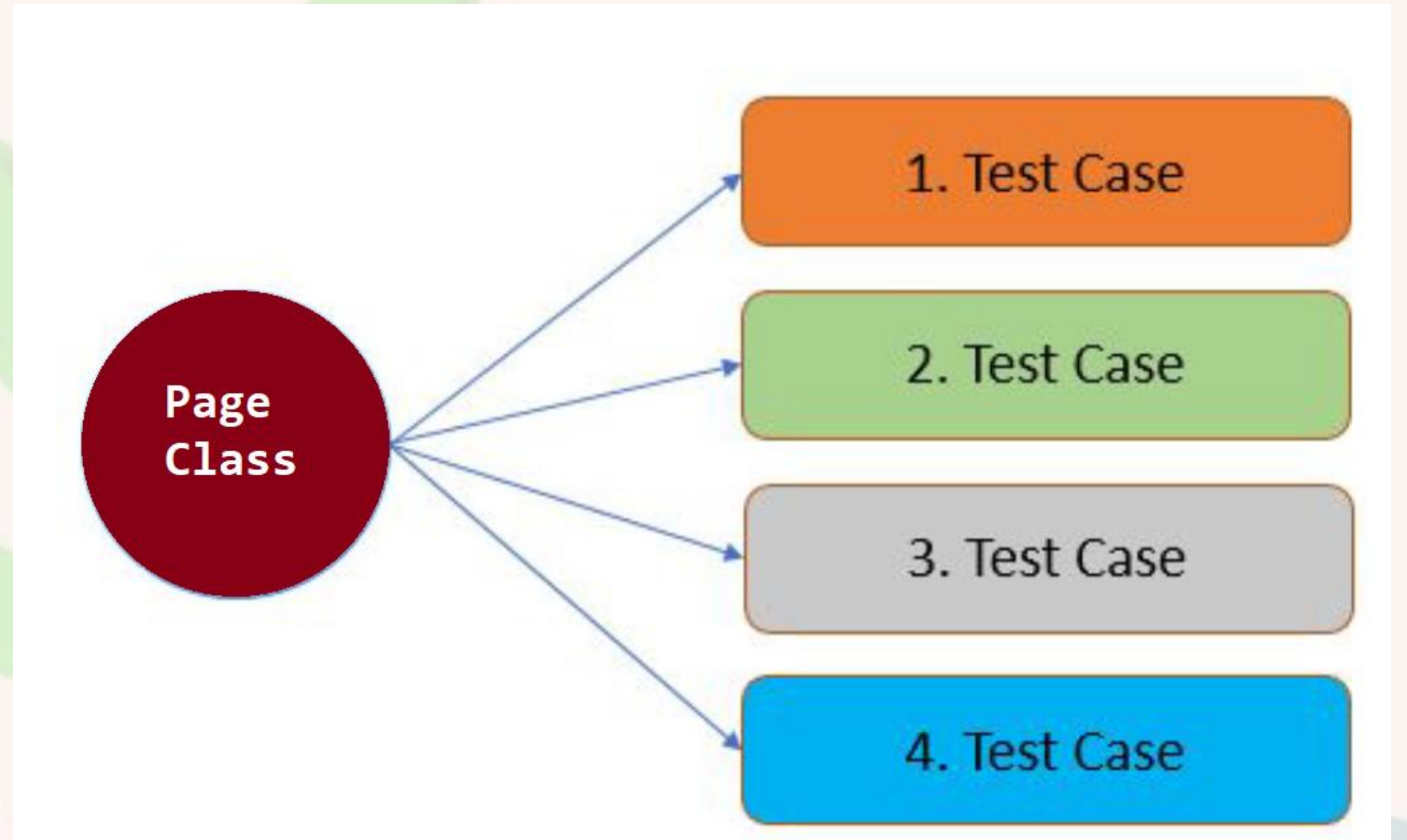
C- Eger kullanacagimiz variable veya method static ise object olusturmadan **direk class ismi ile variable veya method'a ulasabiliriz.**

Okul.variable ile variable'lara

Okul.method() ile methodlara ulasabiliriz.

Page Object Model Framework design

- Daha kullanışlı bir **Framework** oluşturmak için temel hedefimiz, tekrar tekrar yaptığımız işlemleri ve testlerimizde kullandığımız Test Data'larını önceden hazırladığımız dosyalarda tutmaktır.
- Bu şekilde testlerimizde ihtiyaç duyduğumuzda bu verilere kolayca ulaşabilir veya test dataları ile ilgili bir değişiklik yapmamız gerektiğinde sadece kaynak dosyadan bir değeri değiştirerek tüm test case'leri güncellemiş oluruz.



Page Object Model

Framework design

- Bu çok popüler bir **Framework Design Pattern** 'dir.
 - Test suitlerimizde çok fazla testimiz olduğunda, test caseleri ve kodları korumak daha karmaşık hale gelir. Bu nedenle, sürdürülebilir(**maintainable**), yeniden kullanılabilir(**reusable**), daha hızlı(**faster**), anlaşılabilir(**understandable**) daha iyi bir framework dizaynına ihtiyacımız vardır.
 - Page object model ile, sayfaya özgü elementleri veya methodları **page class** içinde tutar, ve bunları gerçek **test class**larından uzak tutarız.
 - Framework un verimliliğini artırmak için core Java ve Selenium konseptini kullanarak temel olarak **page classları** ve **test classları** oluşturacağız.
 - Tüm şirketler page object model dizaynını kullanmaz, ancak herkes bunu bilir ve daha da popüler hale gelmektedir.
- TECH PRO
-

Page Object Model

Framework design

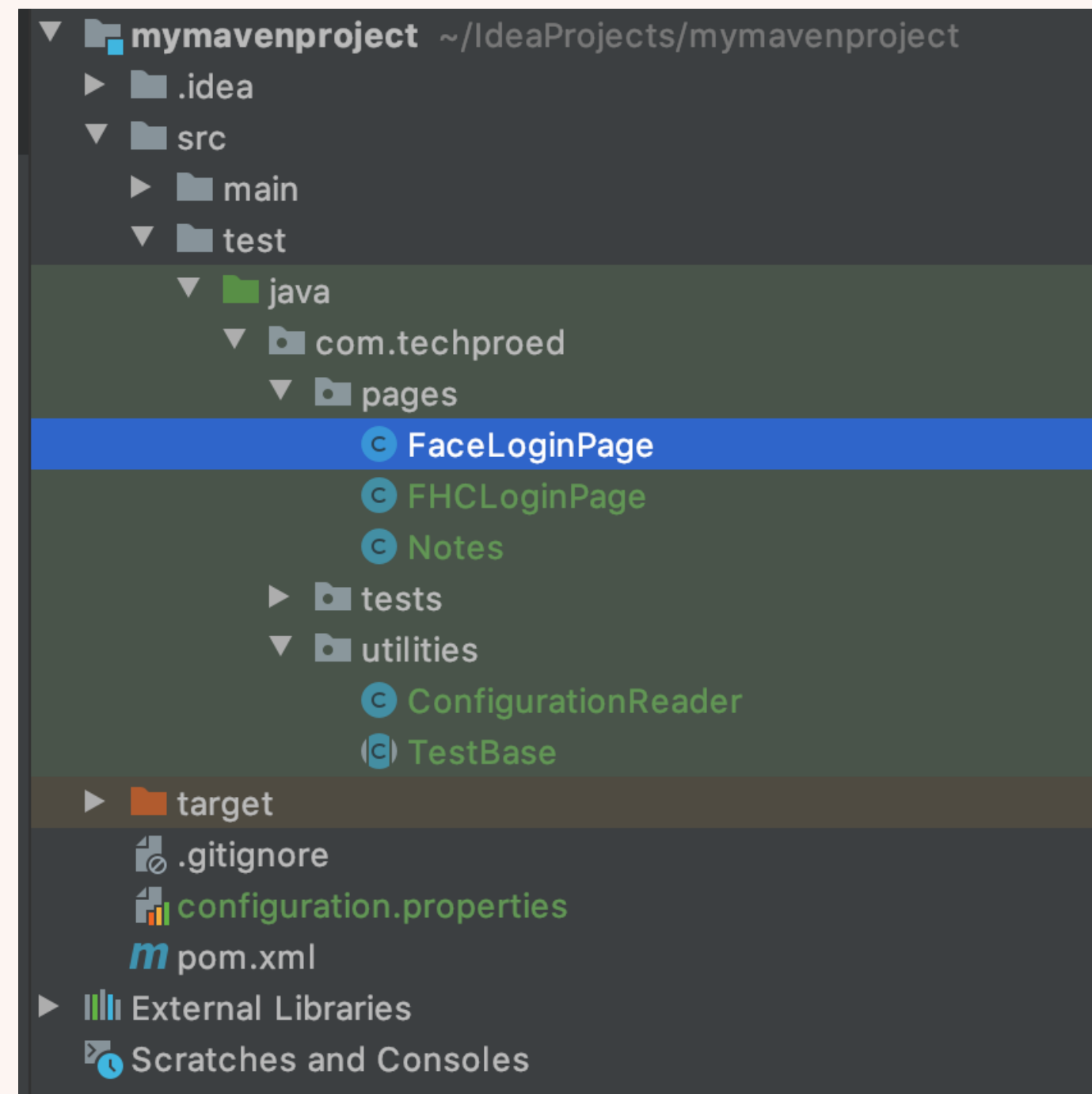
Page Object Model Nedir?

- Test case'leri ve kodları oluştururken Otomasyon framework'un(STLC nin parçası) korunması zaman içinde zor olabilir.
 - Bir uygulamanın(application) işlevselliği değiştiğinde, kodu düzeltmek için framework kontrol etmelidir.
 - Page Object Design daha bağımsız test senaryoları oluşturmamıza yardımcı olur, böylece test komut dosyalarında(script) hata ayıklamak daha kolay olacaktır.
 - **Page Object Design** zorlukları azaltmak amacıyla uygulamaları test etmek için kullanılan bir automation framework design dir.
 - Page Object Design **objectleri, classlari, methodlari, datalari**, etc yeniden kullanabilmeyi mümkün kılar. Böylece testerlar daha az iş yapar.
 - Sadece bir kez create edip birden çok kez kullanıyoruz.
 - Daha iyi bir tasarım, testin yürütme süresini daha hızlı hale getirir.
-

Page Object Model Framework design

Page Object Model temelde 3 package icerir

- Tests : Sadece testleri execute etmek icin gerekli adimlari yazacagimiz class'lar icerir. Hicbir data girisi yapmayacagiz
- Pages : Test yapacagimiz sayfalardaki Web Elementlerini locate etmek ve temel methodlari olusturmak icin kullanilir.
- Utilities : Driver, TestBase ve ConfigurationReader class'larini icerir



Feature: Login feature

Scenario: TC01_kullanici gecerli kimlik bilgileri ile giris yapar

Given kullanıcı fhctriplogin sayfasında

And kullanıcı geçerli username girer

And kullanıcı geçerli password girer

And kullanıcı login butonuna basar.

Then loginin başarılı olduğunu doğrula

Page Object Model Framework design

facebook

Facebook tanıdıklarınla iletişim kurmanı
ve hayatında olup bitenleri paylaşmanı
sağlar.

E-posta veya Telefon Numarası

Şifre

Giriş Yap

[Şifreni mi Unuttun?](#)

Yeni Hesap Oluştur

Ünlü biri, müzik grubu veya şirket için **Sayfa oluştur**.

```
Driver.get("https://www.facebook.com/")
userNameTextBox.sendKeys("mehmet")
passwordTextBox.sendKeys("12345")
girisButonu.click()
```

Page Object Model Framework design

PAGE CLASS

```
public class FaceLoginPage {  
  
    WebDriver driver;  
    public FaceLoginPage(WebDriver driver) { // We need to  
        this.driver = driver; // to connect the page element  
        PageFactory.initElements(driver, this);  
    }  
  
    @FindBy(id = "email")  
    public WebElement username;  
  
    @FindBy(id = "pass")  
    public WebElement password;  
  
    @FindBy(id = "u_0_b")  
    public WebElement loginButton;  
  
    public void login(String userid, String pass) {  
        username.sendKeys(...charSequences: userid);  
        password.sendKeys(...charSequences: pass);  
        loginButton.click();  
    }  
}
```

Face Log In
Page Element

Face Log In
Page Method

TEST CLASS

```
public class LogIn_PageObjectModel extends TestBase {  
  
    @Test  
    public void loginWithPOM() {  
        driver.get("https://www.facebook.com/");  
        FaceLoginPage faceLoginPage = new FaceLoginPage(driver);  
        faceLoginPage.username.sendKeys(...charSequences: "user name");  
        faceLoginPage.password.sendKeys(...charSequences: "password");  
        faceLoginPage.loginButton.click();  
        faceLoginPage.login(userid: "username", pass: "userpassword");  
    }  
}
```


Page Object Model

Page Factory

1. **PageFactory**, page object dizayni için bir önemli classtır.
2. `PageFactory.initElements(driver,this);`
this => page instance
driver => bizim gonderecegimiz driver
3. Page objelerini instantiate(ilk deger atama) için page classlarında **PageFactory** kullanıyoruz.
4. Aslında PageFactory, elementlere ilk degeri atayan **initElements()** metodunu destekler.

```
public class LoginPage {
    WebDriver driver=new ChromeDriver();
    //It is a good practice to put PageFactory in the class constructor.
    //Page factory instantiate the page object class and its elements.
    public LoginPage(){
        PageFactory.initElements(driver, page: this);
    }
    //used to mark an create an element in the page class to be directly called on the test classes.
    @FindBy(id="login")
    public WebElement username;

    @FindBy(id = "password")
    public WebElement password;

    @FindBy(xpath = "//button[@type='submit']")
    public WebElement loginButton;

    public void login(String user,String pass){
        username.sendKeys(user);
        password.sendKeys(pass);
        loginButton.click();
    }
}
```

Page Object Model

@FindBy Annotation

1. **@FindBy** notasyonu test class'larında kullanacağımız Web Elementlerini Page sayfasında locate etmek için kullanılır
2. Bunun için kullanacağımız Web Elementini public olarak oluşturmali, sonra da **@FindBy** notasyonu ile locate etmeliyiz
3. Bu islemi yaptıktan sonra hangi test methodumuzda bu web elemente ihtiyac duyarsak page class'ından uretecegimiz obje uzerinden rahatlikla kullanabiliriz

```
public class LoginPage {
    WebDriver driver=new ChromeDriver();
    //It is a good practice to put PageFactory in the class constructor.
    //Page factory instantiate the page object class and its elements.
    public LoginPage(){
        PageFactory.initElements(driver, page: this);
    }

    //used to mark an create an element in the page class to be directly called on the test classes.
    @FindBy(id="login")
    public WebElement username;

    @FindBy(id = "password")
    public WebElement password;

    @FindBy(xpath = "//button[@type='submit']")
    public WebElement loginButton;

    public void login(String user,String pass){
        username.sendKeys(user);
        password.sendKeys(pass);
        loginButton.click();
    }
}
```

Page Object Model

Class Work

1. <http://qa-environment.crystalkeyhotels.com> sayfasina gidin
2. Smoke testleri POM ile tekrar yapin
3. Positive Login Test
4. Negative Login Test

TECHPROED

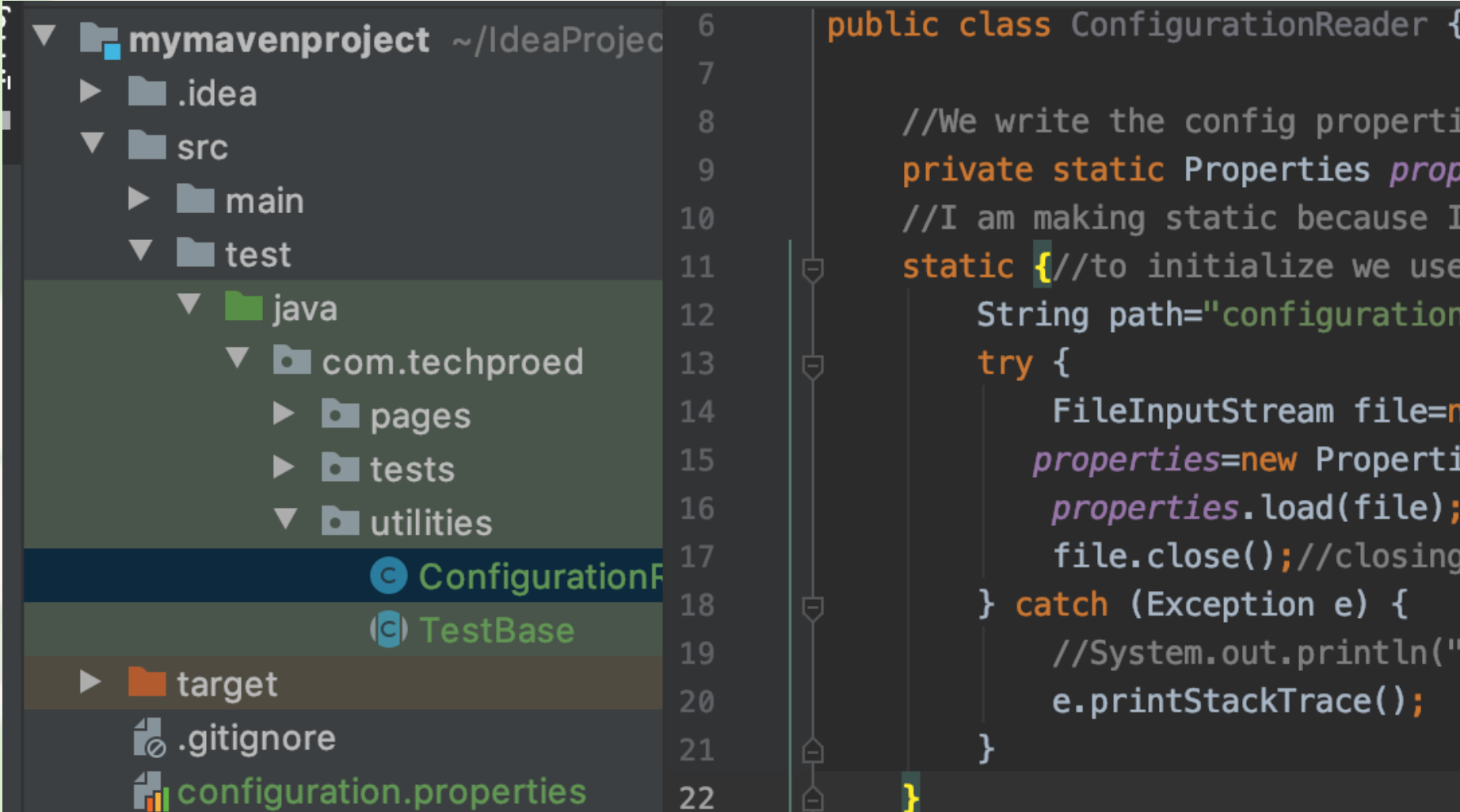
Page Object Model

Properties File

- configuration.properties Test datalarını tuttuğumuz **.properties** uzantılı bir dosyadır.
- Bu dosya , projeyi temiz ve dinamik hale getirir. Test datalarını dinamik hale getirebiliriz.

Örneğin :

driver.get("https://www.google.com") yazmak yerine configuration dosyamıza **url**'i tanımlayıp test classında sadece **driver.get(url)** kullanırız.



```
6 public class ConfigurationReader {
7
8     //We write the config property
9     private static Properties prop
10    //I am making static because I
11    static { //to initialize we use
12        String path="configuration
13    try {
14        FileInputStream file=new
15        properties=new Properti
16        properties.load(file);
17        file.close();//closing
18    } catch (Exception e) {
19        //System.out.println("
20        e.printStackTrace();
21    }
22 }
```

Temel olarak key(anahtar) ve value(değer) çiftlerini kullanırız ve ihtiyaç duyduğumuzda onları çağırırız(url,credentials,browser,environments,...)

- **url**=https://www.fhctrip.com/
- **browser**=chrome
- **username**=manager
- **password**=Man1ager2!
- **name**=Ali

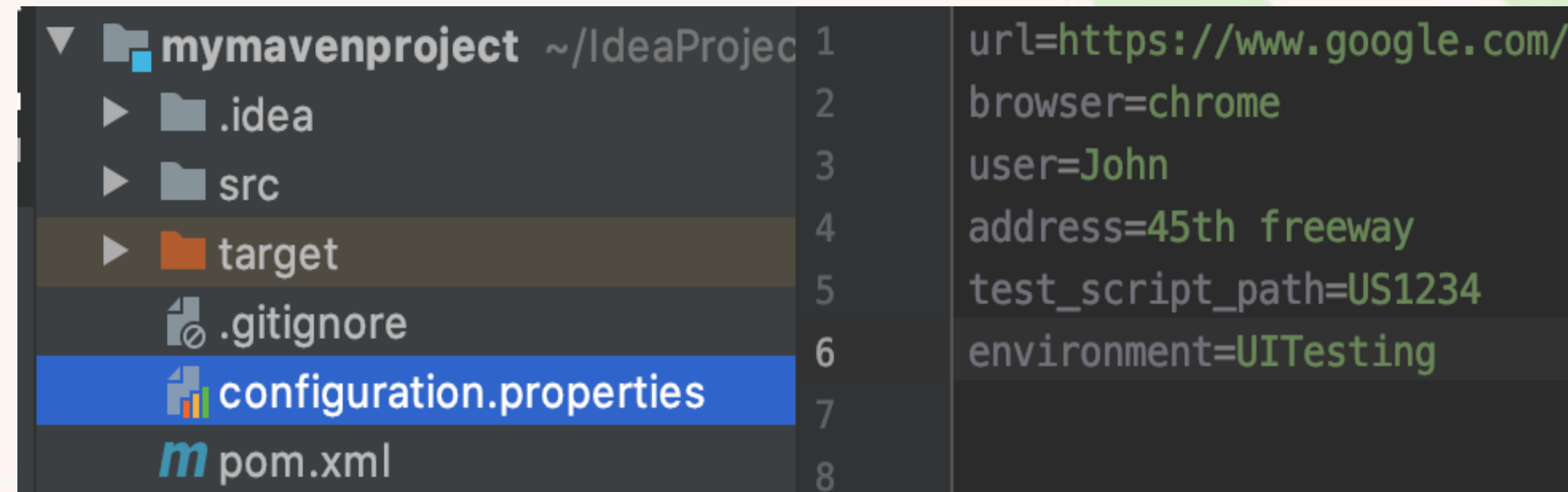
Page Object Model

configuration.properties file

Olusturmak icin project'imize sag click yapin

New->File -> configuration.properties

- properties file test datalari saklar.
- Bu dosyayı kullanmanın amacı, kodu sabit(hard coded) değil, dinamik yapmaktır, böylece herkes verileri kolayca değiştirebilir ve test senaryolarını yürütmek

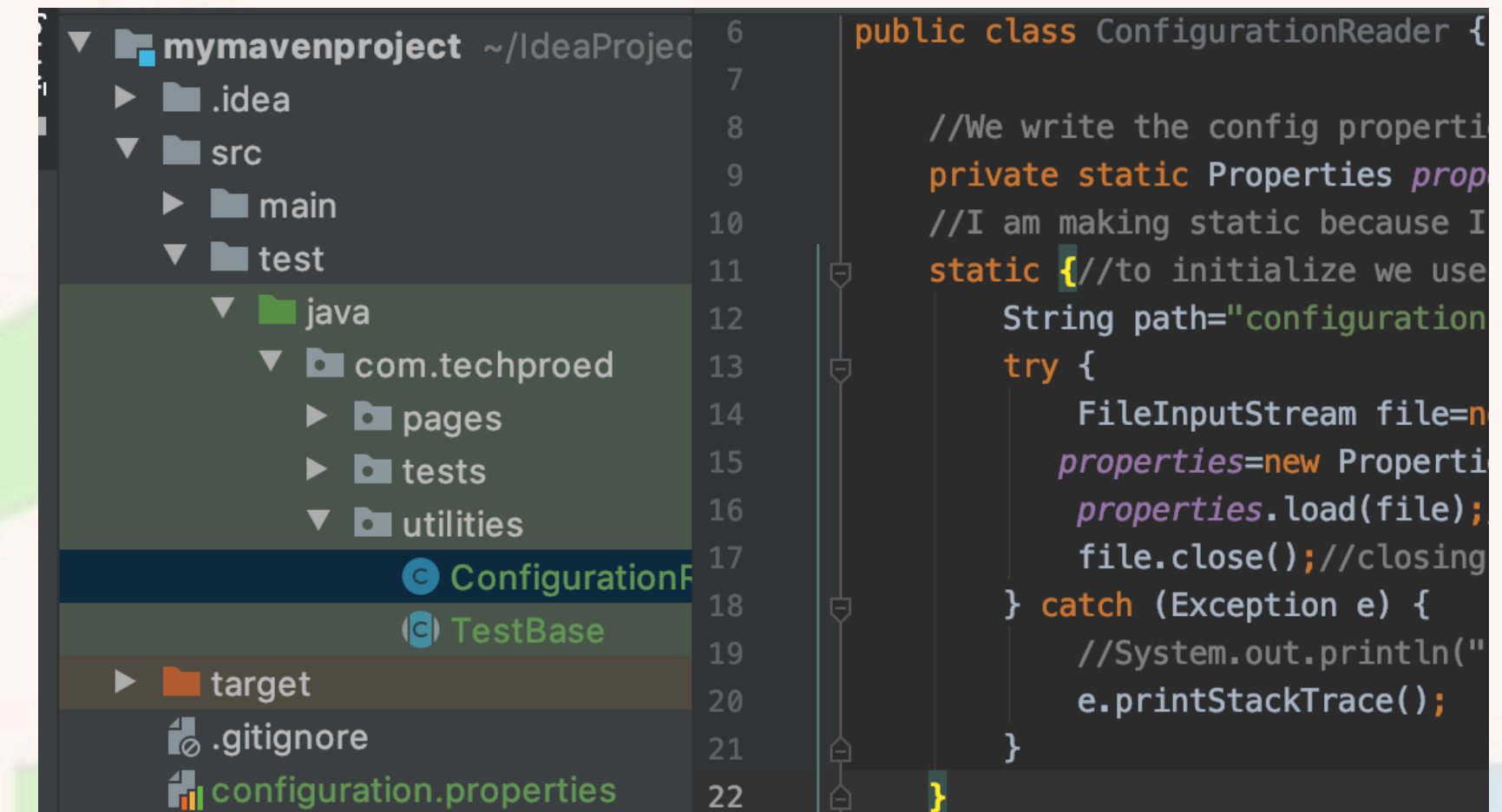


```
1 url=https://www.google.com/
2 browser=chrome
3 user=John
4 address=45th freeway
5 test_script_path=US1234
6 environment=UITesting
7
8
```

ConfigurationReader class

Configuration.properties dosyasından veri okumak için ayrı bir java class'ı oluşturuyoruz.

Bu class'da test class'larından ulaşacağımız **Statik** değişkenler ve bloklar bulunur.



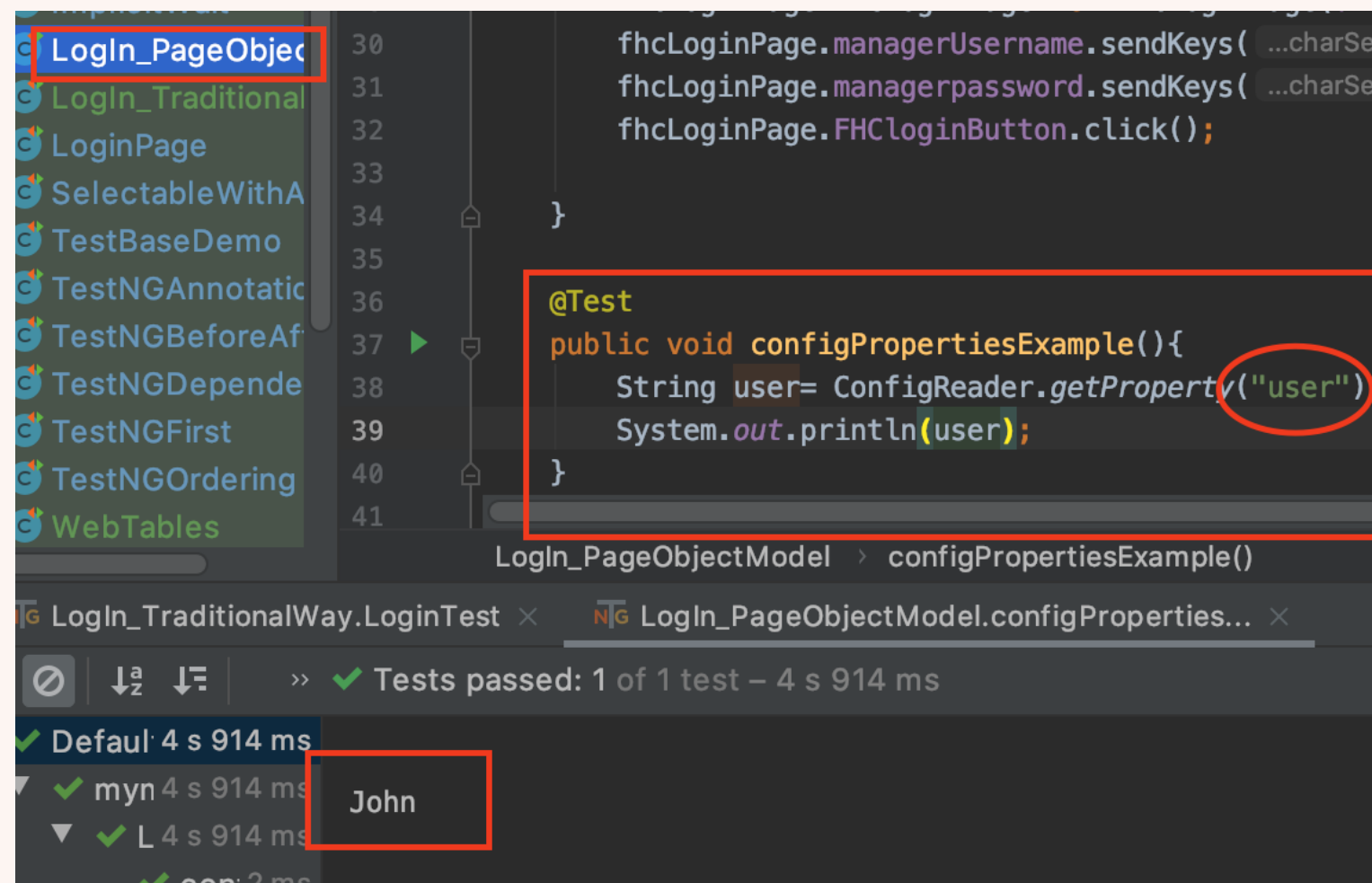
```
6 public class ConfigurationReader {
7
8     //We write the config properti
9     private static Properties prop
10    //I am making static because I
11    static { //to initialize we use
12        String path="configuration
13        try {
14            FileInputStream file=n
15            properties=new Properti
16            properties.load(file);
17            file.close();//closing
18        } catch (Exception e) {
19            //System.out.println("
20            e.printStackTrace();
21        }
22    }
```

Interview Sorusu : Çerçevenizde statik'i nerede kullanıyorsunuz?

Cevap: getProperty yöntemi statiktir, bu yüzden statik olarak kullanabiliriz

Page Object Model Config Reader Data Flow

Test Class



```
30 fhcLoginPage.managerUsername.sendKeys(...charSec
31 fhcLoginPage.managerpassword.sendKeys(...charSec
32 fhcLoginPage.FHCloginButton.click();
33
34 }
35
36
37 @Test
38 public void configPropertiesExample(){
39     String user= ConfigReader.getProperties("user");
40     System.out.println(user);
41 }
```

LogIn_PageObjectModel > configPropertiesExample()

LogIn_TraditionalWay.LoginTest x NLogIn_PageObjectModel.configPropertiesExample() x

Tests passed: 1 of 1 test - 4 s 914 ms

Default 4 s 914 ms

myn 4 s 914 ms

L 4 s 914 ms

con: 2 ms

John

ConfigurationReader Class

```
import java.io.FileInputStream;
import java.util.Properties;

public class ConfigurationReader {

    //We write the config properties one and keep using it.
    private static Properties properties;//We use Properties class to handle properties i
    //I am making static because I want it to run every time I call. I want it to belong
    static {//to initialize we use static block.
        String path="configuration.properties";//path of the config properties file
        try {
            FileInputStream file=new FileInputStream(path);//To open an external file we
            properties=new Properties();//initializing the configurationFile
            properties.load(file);// Loading the file
            file.close();//closing the file after loading. This is not mandatory
        } catch (Exception e) {
            //System.out.println("Path doesn't exist");// we can send a message if file d
            e.printStackTrace();
        }
    }

    public static String getProperties(String key) {
        return properties.getProperty(key);
    }

    //Testing if I can read from the configuration.properties file. Key and Value pairs
    // public static void main(String[] args) {
    //     String a=properties.getProperty("url");
    //     System.out.println(a);
    // }
}
```

Configuration.properties file



```
1 url=https://www.google.com/
2 browser=chrome
3 user=John
4 address=45th freeway
5 test_script_path=US1234
6 environment=UITesting
7
```

Not: Properties dosyasında olmayan bir anahtar(key) alırsak, null değeri döndürür .
Örn: getProperties ("country") -> dosyada ülke yok, bu nedenle null değerini döndürüyor

Page Object Model

Class Work

1. <http://qa-environment.crystalkeyhotels.com> sayfasina gidin
2. Smoke testleri POM ile tekrar yapin
3. Positive Login Test
4. Negative Login Test

TECHPROED
