

# INTERVIEW PREP

# MÜLAKATLARA HAZIRLIK



Batch 81



# Automation Tester ile SDET arasındaki fark nedir?

**Tester:** uygulamayı gereksinimlere göre doğru çalışıp çalışmadığını test eder.

**QA Analyst:** Sistem basından sonuna kadar dahil olup uygulamayı kullanıcı gözüyle test edip raporlarlar.

**Automation Tester:** Hazır üzerinden sadece code güncellemeleri yapar ve hazır sistem benzer kodlar yazarlar.

**QA Engineer:** Kalite kontrol aşamalarını projenin her aşamasında yansıtan ve teknik donanımı güçlü olan kimselerdir.

**SDET (Software Development Engineer in Test):** Sifirdan bir framework un yapımı, raporlarının hazırlanması, dizayn oluşturulması ve yeniden kullanılabilir kodlar yazılması gibi özelliklere sahiptir.



# Buradaki ilk 3 ayında senden ne bekleyebiliriz?

**Eğer kurulu bir düzen üzerinden giden kurumsal bir firma ile görüşüyorsanız....**

**Örnek Senaryo :** Daha önce sizin çalıştığınız domainde çalışmadım. O sebeple öncelikle işin business kısmını kavramalıyım ki, requirementleri daha kolay anlayabileyim.

İlk üç ayımda sizlerin de desteği ile uyum sağlayıp, işi öğrenip, kendi işlerimi sorunsuz bir şekilde yürütebileceğimi düşünüyorum.

**Eğer şirkette çalışacak ilk tester iseniz ve manuel test yapıyorsa...**

**Örnek Senaryo :** Kısa sürede businessi ve gereklilikleri anlayıp, test caseleri yazıp, testleri yürütecek konuma geleceğime inanıyorum.

**Eğer şirkette çalışacak ilk otomasyon tester iseniz....**

**Örnek Senaryo :** En kısa sürede beklentileri karşılayacak şekilde bir framework oluşturup, stabil hale gelmiş kısımların otomasyon kodlarını çalıştırabilir hale geleceğimi düşünüyorum.



# Kendini gelecek 5 yıl içerisinde nerede görüyorsun?

5 yılın sonunda kendimi , kendini zamanın gelişmelerine göre güncellemiş ve en güncel test otomasyon toollarını kullanan, otomasyon testi alanında uzmanlaşmış bir SDET olarak görüyorum.

Özellikler şu an Cypress ve Appium öğrenmek istiyorum. 5 yıl sonra bu toollar hala kullanılır olur mu bilemiyorum ama mobil test alanında kendimi geliştirmek istiyorum.



# PO sana user story verdi. Teste başlamadan önce neler yaparsın?

- Requirementları anlamaya çalışırım.
- Requiementlarda anlayamadığım yerleri netleştirmeye çalışırım.
- Requiriment ve accaptance criteria lara göre test caselerimi oluştururum.
- Test caseleri yazarken olabildiğince kapsayıcı caseler oluşturmaya dikkat ederim.
- Bunun için test tekniklerini uygulamaya çalışırım



## Elinde high seviyede iki task var. Hangisini önceliklendirirsin?

Bu durum tek başıma karar verebileceğim bir şey değil, iki taskin detayını team leade belirttikten sonra hangisini önceliklendirmem gerektiği konusunda fikrini alırım. O en doğru şekilde yönlendirecektir.



# Versiyon nedir? Neden önemlidir?

Yayımlanan yazılımların daha yenileri ve daha eskilerini birbirinden ayırt edebilmek için **versiyon** kategorileme sistemi kullanılır. Versiyon belirleyicisi bir numara olabilirken yalnızca bir kelime veya her ikisi de olabilir. Örneğin 1.0 gibi....

**Kullanılan yazılımın sürüm numarası(versiyonu) neden önemlidir ?**

Kullanıcının kullandığı yazılımın kapsadığı son yenilikler versiyonla anlaşılır.



# Neden Selenium WebDriver tercih ettiniz?

- Open source (açık kaynak ) olduğu için lisans gerektirmez
- Tüm tarayıcılarda çalışır (multi browser testingi destekler)
- Birçok programlama dilini destekler (java,js, python, ruby vb.)
- Farklı işletim sistemlerinde çalışabilir. Platformdan bağımsızdır.
- Kullanımı kolaydır ve çok fazla kaynağa sahiptir.





## Selenium 3 ile 4 arasındaki farklılıkları biliyor musunuz

Araştırılmak üzere şimdilik size bırakıyorum???



# Framework nedir? Framework çeşitleri nelerdir?

Kodun bakımını, kodun okunabilirliğini ve kodun yeniden kullanımını kolaylaştırmaya yardımcı olan bir kod yapısıdır.

Test senaryolarını otomatikleştirmek için kullanılan başlıca 4 tür framework vardır:

## **Data Driven Framework(Data Driven Test- DDT)**

Aynı test senaryosunda birden fazla test datasının kullanabiliriz. Bunun için TestNG de @DataProvider, Cucumber'da ise Scenerio Outline kullanılır

## **Keyword Driven Framework(Keyword Driven Test - KDF)**

Test datalarının harici kaynaklardan alabiliriz. Örneğin Apache Poi aracılığı ile excelden alabiliriz.

## **Behavioral Data Driven (BDD )**

Feature file da oluşturduğumuz steplerdir. Teknik bilgiye sahip olamayan biri dahi test adımlarını rahatlıkla anlayabilir. Ayrıca test adımlarından alınan test datalarını da kolaylıkla değiştirerek, yeni datalarla testi çalıştırabilir

## **Hybrit Data Framework**

Her iki methodunda bir arada kullanılması ile oluşan framework'e denir



## TDD nedir?

Test Driven Design yada Test Driven Development anlamına gelir. Bu yaklaşımda test seneryolarının developer kodunu yazmadan yazılması beklenir. Yani önce test seneryoları oluşturulur. Sonrasında development aşamasına geçilir. Ardından ürün test edilir.

Genel olarak uygulamasının zor olduğu bir yaklaşım olarak kabul edilir.



## pom.xml nedir?

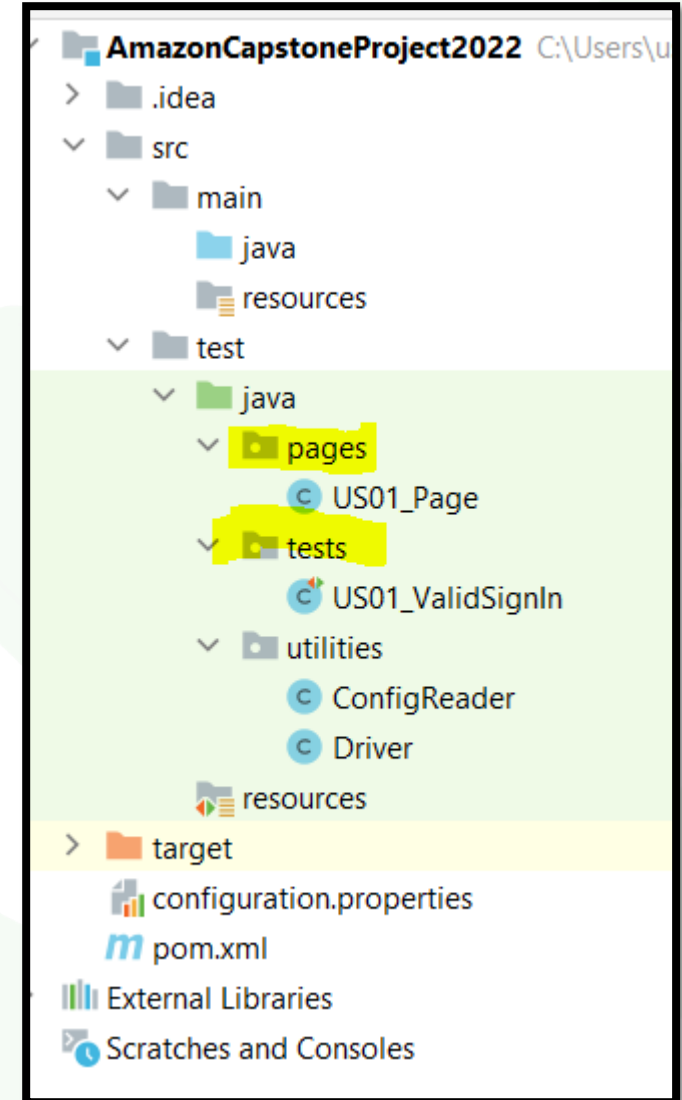
Apache Maven projelerinde karşınıza çıkan POM, xml dosya formatında olan ve bir projenin build edilmesinden, nasıl edileceğine, bağımlılıklarından, packagingine kadar proje ile ilgili olan her türlü bilgiyi içerisinde barındıran bir dosyadır. Maven'in son versionunda pom.xml olarak adlandırılır, önceki versionlarında farklı isimlendirmeler vardı.

POM'un projenin base directory'sinde bulunması gerekir.



# Page Object Model Nedir?

- 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 classlarından uzak tutarız.





Bir web uygulamasında bir web elementin görünmesi bazen kısa bazen uzun zaman alır. Bu durumu nasıl handle edersin?

Bazen otomasyon kodlarımız ile web tarayıcımız aynı hızda çalışmadığı için senkronizasyon sorunu yaşarız. Farklı sebeplerden ötürü (internet hızımız, bilgisayarımızın kapasitesi vb. ) web elementleri bazen çabuk bazen de daha yavaş yüklenebilir. Bu gibi durumlarda yaşanacak sorunları ortadan kaldırmak için **waitleri** kullanabiliriz.



# Seleniumda wait çeşitlerinden bahseder misin?

## 4 çeşit wait kullanırız

**Thread.sleep()** → javadan gelir. Hard wait . Belirlediğimiz süre kadar kodu bekletir.

**Explicitly wait** → spesifik bir web elementi için kullanılır. Kullanırken obje üreterek kullanıyoruz. Belli bir şarta göre bekleme yapar. Örneğin web elementi tıklanabilir olması gibi, şart gerçekleştiğinde beklemeyi durduruyor.

**implicitly wait** → Bir sayfadaki tüm öğeler için belirli bir süre sürücü (driver) bekler. Tester'lar genellikle otomasyon frameworklerinde sürücü nesnesini (driver object) varsayılan olarak implicitly wait ile bekletir. Web elementi bulunduğu bekleme durur.

**Fluentwait** → bu wait obje oluşturularak kullanılır. Döngüsel bir zaman diliminde bir web elementini bulana kadar için tekrarlar.



# Locaterlardan bahseder misiniz?

8 tane locate yöntemi var.

id,name,className,tagName, xpath, partialLinkText,Css selector, linkText

İlk tercih edilmesi gereken id ve name dir.

<a > -- linktext,partialLinktext

## Hangi durumlarda web elementlerini bulamayız

- Locate değişmiştir
- Farklı iframe de bulunması
- Senkronizasyondaki sıkıntılar
- Scrool yapmak gerektiği durumlarda görünmeyebilir.
- Sayfanın tam yüklenmediği durumlar
- Html kodunun içerisinde js ile oluşturulan web elementlerinde..
- Driver farklı bir tabda olabilir(windows handle)





# Xpath çeşitlerinden bahseder misin?

iki çeşit xpath var

→ Absolute -- `/../..` Parent child ilişkisini atlamadan yazıyoruz  
`/div/div/a/...../..aaaa`

- locate zor olur...
- sayfaya eklemeler yapıldığında artık işlevsizdir, değişir

→ Relative -- `//` web elementine doğrudan ulaşır. Parent child ilişkisine gerek yok.

`//div/div/a...`

`(//div[@class='xyt'])[5]`



# Dinamik bir web elementini nasıl locate edersin?

Dinamik web elementi ne demektir?

Farklı zamanlarda yeri değişebilen web elementleridir. Örneğin bir text alanı olduğunu düşünün. Bu text alanının yeri belirli aralıklarla değişebilir.

Dinamik web elementleri bulunurken css selector ve ya xpath kullanılabilir.

- `contains( ) -- //*[contains(@name='btn')]`
- `startswith( ) -- //label[startswith(@id, 'message')]`
- `text( ) -- //td[text() = 'usedId'] //*[text()='a']`
- `or & and -- //input[@type = 'submit' and @name = 'login']`