

ÇERÇEVENİZİ SÖYLEYİN

1. Bana çerçevenizden bahseder misiniz?

Versiyon 1

- Son projemde çerçevemi tasarlamak için **Eclipse IDE** olarak farklı yönetim ve otomasyon araçları kullandım /
IntelliJ , **Tarayıcı otomasyonu** için **Selenium WebDriver** , bağımlılıklar için **Maven** , **Cucumber** ve **Jenkins** . Ben de kullandım
 Kodumu düzenli ve temiz tutmak için **POM** yapısı. Bu nedenle, temelde her sayfa için ayrı bir **Java sınıfı** oluşturdum
 o sayfanın tüm öğelerini ve ilgili yöntemleri depoladığım uygulamamın Ayrı sınıflarım var
 uygulanan **adım tanımlarını koruyun** . Ayrıca, her senaryo için (pozitif veya negatif) **Salatalık özellik dosyaları** oluşturdum
Test senaryolarımı açıklamak için **GHERKIN** dilini kullandığımda , bunu yaparak test durumlarımın
 teknik olarak güçlü olmayanlar için bile ekibimin her bir üyesi için anlaşılabilir. Başka bir ayrı yarattım
 tüm sürücülerimi, sayfa dosyalarımı veya çalıştırabileceğim yardımcı programları saklayacağım yardımcı programlarım için bir paket. Raporlama yapıldı
 salatalık ve Jenkins'te. Aslında, Jenkins test her çalıştırıldığında salatalık tarafından oluşturulan Json dosyasını kullanıyor.
 Tabii ki, tüm kodum GIT'de saklanıyor, böylece gerektiğinde ekip üyelerimle paylaşabilirim. Seveceğim son şey
 Ek olarak, GIT'den (Sürüm Kontrolü) kodu alarak duman ve regresyon testlerimi çalıştırmak için Jenkins kullanıyorum.

Versiyon 2

- Bağımlılıklarımı ve eklentilerimi **pom.xml'de yönetiyorum** , testleri komut satırından çalıştırıyorum. **Java** ile yazılmıştır . kullanırım
Eclipse IDE / IntelliJ şirketimde. Benim çerçeve kullanan **Selenium WebDriver** için **tarayıcı otomasyon** ve **JUnit** için
 Benim testleri başlayan ve iddialar için, My çerçevesidir **BDD** kullanır, **Salatalık** özelliği dosyasında yazma testlere organize
 @smoke, @api gibi test paketleri. Özellik dosyalarımız, **olmayanların** anlaşılmasını kolaylaştırmak için **Gherkin** dilinde yazılmıştır.
 teknik insanlar. Çerçevem, özellik dosyasından tam adıyla **HTML** raporları oluşturur .
 Benim çerçevemde; Özellik dosyaları dışında, testlerimi çalıştıran ve adım için kodlar oluşturmaya yardımcı olan bir koşucu sınıfım var.
 özellik dosyalarından tanım. Ayrıca, özellik dosyalarımın ve **StepDefs'in** (Glue) nerede olduğunu gösteren konumları da içerir .
 Özellik dosyalarını yürütmek için yöntemlerimin bulunduğu adım tanımlama sınıfları, Driver sınıfı **Singleton** olarak tasarlanmıştır .
 Yapılandırma Okuyucu ve özellikler dosyası. Ayrıca her sayfa için öğelerimi bulduğum sayfa sınıflarım var. dışında
 bunlar, **yeniden kullanılabilir kodlarımı** sakladığım yardımcı program sayfalarım var . Son olarak, kodlarımı uygulayan kanca sınıfım var.
 tüm testlerimden önce ve sonra çalıştır , her hata için **TakeScreenShot'ımı** çalıştırdığım yer **burasıdır** . Çerçevem destekler
Apache POI , **Scenario Outline**, **excel ve csv dosyalarımı** kullanarak **Veriye Dayalı test** .
 Benim çerçevemde, JDBC sürücüsü aracılığıyla Veritabanı testi de yapabilirim. **JIRA'yı** proje yönetimi olarak kullanıyorum ve
 hata izleme aracı. Sürekli Entegrasyon ve test planlaması için **Jenkins kullanıyorum** . Ve çoklu kullanım için **Selenium Grid** kullanıyorum
 tarayıcı paralel testi.

2. Çerçevemin özeti?

ÇERÇEVE

Veriye Dayalı Çerçeve

Testlerin veri setine dayalı olarak yürütüldüğü bir çerçeve, çerçeve gibi dış kaynaklardan gelen verileri okumak için tasarlanmıştır.
 verilere göre testler yapın ve çalıştırın. Veriye dayalı çerçevede, aynı testi birden çok kez gerçekleştirebiliriz.
 farklı veri kümeleri.

Sayfa Nesne Modeli Çerçevesi

Uygulamanın her sayfası için ayrı bir java sınıfı oluşturduğumuz sayfa nesnesi tasarım modelini kullanır

Davranış Odaklı Çerçeve

Anahtar Kelime Odaklı Çerçeve

KDF'de, testlerimizi çalıştırmak için dış kaynaktaki (excel. Csv) anahtar kelimeleri kullanırız. Çerçeve, verileri ve adlarını okumak için tasarlanmıştır
 excel'den alın ve buna göre eylemler yürütün.
 KDF kurulduktan sonra, teknik olmayan test uzmanları bile otomatik testler yazabilir ve yürütebilir.

- 10 -

Hibrit Çerçeve

hibrit çerçeve, yukarıda verilen türlerden en az ikisini kullanan bir çerçevedir

ARAÇLAR

Java : Çerçevem Java dili kullanılarak yazılmıştır

Maven : Bir maven projesi olarak oluşturulan çerçevem, maven bağımlılıkları yönetmek ve ayrıca testlerimizi mvn olarak çalıştırmak için kullanılıyor
 terminalden hedefler

Selenium WebDriver : tarayıcıyı otomatikleştirmek için kullanılan bir kitaplık / araç / api, tarayıcıyla etkileşim kurar.

TestNG : xml dosyalarını kullanarak testleri gruplamak, yumuşak ve zor iddialar yapmak, test yöntemleri oluşturmak, belirli bir sırayla çalıştırmak için kullanılır
Kapsam : benim çerçevem, teknik olmayan ekip tarafından okunması ve anlaşılması kolay olan ayrıntılı HTML raporları oluşturur
 üyeler. Raporlarım, meydana gelebilecek arızalar için ayrıntılı test adımlarını ve ekran görüntülerini içerir. Ayrıca neye ilişkin metrikler de yapabilir

yüzde geçiyor, başarısız oluyor, atlanıyor vb.

IDE : Mevcut çerçevede IntelliJ kullanıyorum, ancak daha önce kullandığım Eclipse ile de oldukça rahatım

TASARIM

Sayfa Nesne modeli : çerçevem, sayfaların sayfaları için ayrı bir sınıf oluşturduğum sayfa nesnesi modelini kullandı. benim başvurum.

PageFactory tasarımı : Sayfa nesnesi sınıfına erişimi kolaylaştıran bir tasarım.

Sayfa fabrika tasarımının altında değil. PageFactory tasarımıyla aynı adı taşıyan sınıftır:

```
PageFactory.initElements (sürücü, bu)
```

Singleton Driver : Çerçevelerim, webdriver örneğini farklı sınıflar arasında paylaşmak için tekli bir model kullanıyor

TestBase : Benim çerçevem, testlerimin kapsamına giren bir testbase sınıfına sahip. testbase sınıfı, tüm testlerim için ortak adımlara sahiptir.

Configuration.feature dosyası : önemli test verilerini saklamak için kullanılır

Yardımcı programlar: çerçevemin farklı sınıflarında kullanılabilen yeniden kullanılabilir yardımcı programlara sahip

Faydaları :

Bakımı kolay:

- Çerçevem, bakımı kolaylaştıran sayfa nesne modeli kullanıyor. örneğin, herhangi birini güncellemem gerekirse bulucu, sadece bir kod değişikliği yapmam gerekiyor.
- Testlerimi birbirinden bağımsız yapmaya çalışırım. bu, bir testi güncellersem, diğerlerini etkilemeyeceği ve ayrıca biri başarısız olursa, diğerleri etkilenmeyecektir.

Genişletmesi kolay:

- Çerçeveme yeni test senaryoları eklemek kolaydır.

Yeniden kullanımı kolay:

- Tüm testler için yeniden kullanılabileceğim sayfa nesne modeline, yardımcı programlara sahibim.

Çoklu tarayıcı testi : Çerçevem, minimum kod değişikliğiyle farklı tarayıcılarda aynı testleri çalıştırabilir. Junit, TestNG, çoklu tarayıcı testi için kendi yöntemlerine sahiptir. Onlar dışında Selenium Grid kullanıyoruz

Test türleri : Çerçevem, uygulamanın kullanıcı arayüzünü, veritabanını ve API'sini test edebilir.

- 11 -

Sayfa 12

Paketleme : Farklı sınıf ve mantık türleri için farklı paketler oluşturdum. Her sayfa paketi yalnızca sınıfları içerir aynı işlevsellğe sahip.

Adlandırma kuralları : **Ekibimde** kodlama standartlarına, özellikle adlandırma kurallarına çok dikkat ediyoruz. Sınıflar, yöntemler değişkeni, yaptıkları işe göre adlandırılır ve bir standardı izler

Sayfa nesnesi sınıfı :

ana sayfa, giriş sayfası

değişken :

loginButton, signOutLink

yöntemler :

login () : bu yöntemler yalnızca oturum açmak için kullanılır, diğer işlevler için kullanılmaz.

3. Ürün Yapısı? Web Uygulama Yapısı

- Başlangıç aşaması
 - o **SmartClient** Framework → San Francisco tabanlı bir istemciden bir **JavaScript (Js)** UI Framework'tür. % 100 Microsoft uyumlu
 - o Terminoloji
 - **HTML yapısı** : yalnızca sorumlu verilerdir
 - **CSS sunumu / görünümü** : tasarım için
 - **JavaScript dinamik / eylemi** : işlev için
- Arka Uç → **ServerSide Projesi**
 - o Tüm dünyada pek çok **ürünümüz** (makinemiz) var. Özellikle ABD, Avustralya, Singapur, Fransa'da
 - o Her makinede üç sunucu açıyoruz
 - **Uygulama Sunucusu**
 - **IIS Uygulaması** → Kodun -kulaklı görevleri çalıştırdığı mimari. İçin genel URL adresleri veriyoruz her müşteri. Arka uç ve Ön uç projeleri, bu Uygulama sunucularında **IIS Uygulaması** tarafından çalıştırılır .
 - Test çerçevemiz de bu sunucuda barındırılıyor.
 - Geliştirici ekibimiz uygulamaları geliştirmek için C ++ kullanıyor ve biz testçiler Testing projesi için **JAVA** kullanıyoruz
 - **Rapor Sunucusu**
 - Bu, **Windows SQL Hizmetinin** bir hizmetidir .
 - Arayüzden Raporlar oluşturan ve çağıran Rapor hizmetlerimiz mevcuttur.
 - **Microsoft SQL Server** → SQL tablolarımız bu sunuculardadır. Verileri Microsoft SQL Server'da barındırıyoruz.

4. Mobil Uygulama Yapısı

- **Android Uygulamaları** → **Uygulamaları** Android Studio üzerinde geliştiriyoruz. Ve Push gibi birçok AWS hizmetini kullanıyoruz
Bildirim, Mesajlaşma ve E-posta Gönderme
- **IOS Uygulamaları** → XCode 9 üzerinde geliştiriyoruz
- **FireBase** → Mobil Uygulama kullanımını, hataları ve çökmeleri izlemek için (Crashlytics)
- **Appium** → Mobil uygulamaları test etmek için (yeni başladık)
- **Selendroid** → **Gözlemliyor** ve kullanmayı düşünüyoruz.

- 12 -

Sayfa 13**BİR DEĞERLENDİRMEDE SORULACAK BÜYÜK SORULAR**

Kendi kendinize sormak için bazı sorular sormak önemlidir. Pozisyon hakkında ne bilmek istersiniz? Şirket? The Bölüm? Takım? Öyleyse, bize sorunuz var mı? röportajın bir kısmı geliyor mu? Emin olmak için bu listeyi kullanın tüm üslerinizi kapladı.

1-11 İş Hakkında Sorulacak Sorular

Öncelikle, işin günlük sorumluluklarının tam olarak ne olacağından emin olun - hem şimdi hem de gelecek.

1. Tipik bir gün nasıl görünür?
2. Ele alınması gereken en acil projeler nelerdir?
3. Üzerinde çalışacağım projelerden örnekler gösterebilir misiniz?
4. İdeal bir adayda aradığınız beceriler ve deneyimler nelerdir?
5. Bu pozisyonda gerçekten başarılı olmak için birinin sahip olması gereken özellikler nelerdir?
6. Yeni bir işe almak istediğiniz takımda ne tür beceriler eksik?
7. Bu pozisyondaki birinin karşılaştığı en büyük zorluklar nelerdir?
8. Çalışma saatleri ve fazla mesai beklentileriniz nelerdir?
9. Bu oluşturulmuş yeni bir rol mü?
10. Bu yeni bir rol mü yoksa işte daha önce birisi var mıydı?
11. Rol daha önce doldurulmuşsa, o kişi neden devam etti?
12. Bu pozisyonun temel sorumluluklarının önümüzdeki altı aydan bir yıla kadar değişmesini bekliyor musunuz?

Eğitim ve Mesleki Gelişim Hakkında Sorulacak 13-20 Soru

Her yeni işi sadece bir iş olarak değil, kariyerinizde başarıya giden yolda bir sonraki adım olarak düşünün. Bu pozisyon oraya ulaşmanıza yardımcı oluyor mu?

13. Nasıl eğitim alacağım?
14. Yeni işe aldığımız kişi ne tür bir eğitim alacak?
15. Çalışanlarınız için hangi eğitim programları mevcut?
16. İlerleme veya mesleki gelişim için fırsatlar var mı?
17. Şirket, çalışanlarına herhangi bir mesleki gelişim fırsatı sunuyor mu?
18. Şirketi endüstri konferanslarında temsil edebilir miyim?
19. Bu işi yapan son kişi nereye gidiyor?
20. Bu pozisyondaki başarılı çalışanlar daha önce nereye ilerledi?

Performansınız Hakkında Sorulacak 21-25 Soru

Potansiyel yeni yöneticinizin başarılarınızı nasıl ölçeceğini anlamak, hem şirketi anlamamanın anahtarıdır. öncelikleri ve yönetim tarzları.

21. İşteki ilk 30, 60 ve 90 günde birinin beşardığını görmek isteyeceğiniz en önemli şeyler nelerdir?
22. Bu pozisyonun ilk 12 ayda performans beklentileri nelerdir?
23. Çalışanlar performansları hakkında nasıl geri bildirim alıyor?
24. Buradaki gibi performans inceleme süreci nasıldır? Ne sıklıkla resmi olarak incelenir?
25. Performansım hangi ölçütlere veya hedeflere göre değerlendirilecek?

26-30 Görüşmecisi Hakkında Sorulacak Sorular

Görüşmeciyi sorular sormak, onlarla bir kişi olarak ilgilendiğinizi gösterir ve bu, geliştirmenin harika bir yoludur. uyum.

26. Şirkette ne kadar süredir çalışıyorsunuz?
27. Buraya geldiğinizden beri rolünüz değişti mi?
28. Bundan önce ne yaptınız?
29. Bu şirkete neden geldiniz?
30. Burada çalışmanın en sevdiğiniz yanı nedir?