
Sayfa 132

MARUFJAN Sınıflarından

Jenkins nedir?

- Sürekli entegrasyon aracı.
- Geliştiriciler ve testçiler tarafından kullanılan derlemeleri, dağıtımları vb. Planlamak ve otomatikleştirmek için kullanılır.

Jenkins kullanılarak hangi tür işlemler otomatikleştirilebilir?

- birim testleri çalıştırma
- uygulamayı oluşturma
- otomatik kullanıcı arayüzü testleri çalıştırma
- farklı ortamlara dağıtım

Şirketinizde jenkins test için nasıl kullanılıyor?

- Projemde, otomatik duman testlerini planlamak ve yürütmek için jenkins kullanıyoruz.
- duman testlerimiz her gün yapılmaktadır.

Jenkins üzerinde hangi testlerin var?

- Projemde jenkins için duman ve regresyon testlerimiz var.

Jenkins'te ne tür testler yapılabilir?

- Böylece, jenkins herhangi bir otomatik testi çalıştırabilir.
- Örneğin, birim testleri, duman testleri, entegrasyon testleri, regresyon testleri, mantıklı olabiliriz.

Jenkins kullanılarak hangi test katmanları test edilebilir?

- Uygulamanın farklı katmanlarının test edilmesi test kodumuz ile yapılır.
- Jenkins, kullanıcı arayüzünü veya veritabanını veya api'yi test etmemizi önemsemez. sadece testleri başlatır ve raporlar gönderir.
- Dolayısıyla, otomatik testim bir UI testiye, bu, jenkins'in UI testleri çalıştırdığı anlamına gelir. veya otomatik testim bir API testiye, jenkins, API testleri çalıştırıyor.

Jenkins'i kim kurar?

- Şirketimde, jenkins'in bakımından sorumlu DevOps / Operasyon Destek ekibimiz var. kurulu olanlar ve yapılandırılmış jenkins.
- Konuşlandırmalar vb. İçin, uygulamaları oluşturmak ve dağıtmak için işler yaratmak üzere geliştiricilerle birlikte çalışırlar.
- Duman testleri oluşturmak için otomasyon mühendisleriyle birlikte çalışırlar.

Jenkins'te duman testlerini kim kurar?

- Bir otomasyon mühendisi olarak duman testlerimi oluşturmak için devop'lar / operasyon destekleri ile çalışıyorum.
- Testlerimi çalıştırmak için Jenkins üzerinde kurulu bazı eklentilere ihtiyacım var, ayrıca javaya, jenkins üzerinde yapılandırılmış maven'e ihtiyacım var, ayrıca testlerin çalışacağı sunucuda yüklü tarayıcılar.
- Projemde, yalnızca devops / operasyon destek ekibi üyeleri yukarıdaki yapılandırmaları yapma hakkına sahip. bu yüzden sahibim onlarla çalışmak için.
- Jenkins'in yapılandırması tamamlandığında duman testi işini oluşturabilir ve çalıştırabilirim.

Jenkins üzerinde duman testleriniz nasıl yapılandırılıyor?

- Jenkins işimiz, otomatik testlerimi GitHub'dan alacak ve her gün maven kullanarak çalıştırılacak şekilde yapılandırıldı.

- yapılandırmada önce yeni bir iş oluşturdum ve "duman testleri" adını verdim. sonra Kaynak Kodda git seçeneğini seçtim
Yönetim bölümüne gidin ve GitHub'daki çerçevemin yolunu girdim. ve ayrıca git kimlik bilgilerini girdi.
- bir sonraki yapılandırma, düzenli olarak oluştur seçeneğini belirlediğim ve testlerin ne sıklıkla çalıştırılacağını girdiğim tetikleyicilerle ilgili.

Sayfa 133

Build bölümünde , otomasyon çerçevem maven kullanılarak oluşturulduğundan, üst düzey maven komutunu çağır seçeneğini seçiyorum.

- bölümünde maven komutunu giriyorum (mvn kısmı olmadan). bu yüzden testimi terminal kullanarak çalıştırmak için normal komut
'mvn test', jenkins'e sadece 'test' giriyorum. burada komut satırını kullanarak çalıştırmak istediğim etiketi de giriyorum.

test -Dcucumber.options = "- etiketler @smoke"

- regresyon testleri çalıştırmak istersem, komut şu şekildedir:

test -Dcucumber.options = "- etiketler @regression"

Derleme Sonrası İşlemlerde , çift yapılandırma yapıyorum,

- Salatalık raporları seçeneği ekliyorum. Bu kullanılabilir çünkü bizim jenkins'imize salatalık html raporları eklentisi yükledik.
Salatalık raporları eklentisi, her derleme için html raporu oluşturacaktır. Jenkins üzerinde her duman testi yaptığımızda, yeni bir bildiri. ve tüm derleme için tüm raporlar kaydedilir.
- Sonraki Oluşturma Sonrası Eylemler olarak e-posta seçeneği ekliyorum. Burada bunu, Agile ekibimdeki herkesin hakkında bilgilendirilmesi için yapılandırıyorum.
test sonuçları.

Testler için e-posta raporlarını kim alır?

- Agile ekibimdeki herkes test sonuçları hakkında bilgilendirilir.

"DevOps ardışık düzeninde çalıştınız mı?"

- Devops boru hattının parçası olan jenkins üzerinde duman testleri yaptım. devops ardışık düzeni tarafından oluşturulur ve yönetilir
DevOps / İşlem Desteği. Ancak duman testimiz boru hattının bir parçası.
- Ben de jenkins üzerinde duman testleri oluşturup yapılandırarak katıldım.

Entegrasyon testi ile uçtan uca test arasındaki fark nedir?

- Google genellikle 70/20/10 ayrımı önerir:% 70 birim testleri,% 20 entegrasyon testleri ve% 10 uçtan uca testler.

• Birim Testleri

- o En küçük işlevsellik birimini, tipik olarak bir yöntem / işlevi test eder (örneğin, belirli bir duruma sahip bir sınıf verildiğinde, x sınıfındaki yöntem y'nin olmasına neden olmalıdır). Birim testleri belirli bir özelliğe odaklanmalıdır

• Entegrasyon Testleri

- o Entegrasyon testleri, kod birimlerini birleştirerek ve ortaya çıkan kombinasyonun doğru çalışıyor. Bu, bir sistemin içgüdüleri olabilir veya birden fazla sistemi bir araya getirerek yapılabilir.
kullanışlı bir şey. Beyaz kutu test yaklaşımıdır. Geliştirici tek tek birim geliştirir ve bunu entegre eder birbirinizi ve test edin.

• Fonksiyonel Testler

- o Fonksiyonel testler, belirli bir girdiye ait sonuçları,
Şartname. Fonksiyonel testler ara sonuçlarla veya yan etkilerle ilgilenmez, sadece sonuçla (x'i yaptıktan sonra y nesnesinin z) durumuna sahip olması umrunda değil.

• Kabul Testi

- o Bu, yazılım müşteriye teslim edilmeden önce gerçekleştirilen son testtir. Sağlamak için yapılır.
geliştirilen yazılım tüm müşteri gereksinimlerini karşılar. İki tür kabul testi vardır - biri dahili kabul testi (Alfa testi) olarak bilinen geliştirme ekibinin üyeleri tarafından gerçekleştirilir ve (Beta testi) olarak bilinen müşteri veya son kullanıcı tarafından gerçekleştirilen diğer

• Uçtan Uca test

- o, a noktasından z'ye gittiğiniz yerdir ve oraya giderken çeşitli farklı noktalara dokunursunuz. Bir tek için olabilir
bir e-posta gönderme işlemi gibi bir sistem veya birden fazla sistemin dahil olduğu durumlarda kullanılabilir, örneğin öğrencinin bir sınava kaydolması, sınava girmesi ve sonunda puanlarını alması.
- o Gerçek dünya sistem testi. Uygulama tüm entegre donanım, veritabanı, ağ ve diğer arayüzler.

Sayfa 134

Bunları Jenkins üzerinde nasıl çalıştırıyorsunuz?

Uzun hikaye...

Bana çerçevenizden bahseder misiniz?

- En son Book-IT otomasyon çerçevemde, otomatik testler için Cucumber BDD çerçevesini kullandık. bu çerçeve çok esnek bir çerçeve. Birçok farklı otomasyon testi konseptini entegre eden karma bir çerçevedir.
- Çerçeve, MAVEN kullanılarak oluşturulmuştur. maven, çerçeveyi oluşturmak, bağımlılıkları ve eklentileri yönetmek, çalıştırmak için kullanılır.
maven yaşam döngüsü olarak testler.

- Java dili kullanılarak yazılmıştır.
- Bir Hiyar BDD çerçevesidir. Salatalık, otomatik testi teknik olmayanlar tarafından anlaşılır hale getirmek için kullanılan bir araçtır.
 - takım üyeleri. Salatalık, otomasyon mühendisleri ile teknik olmayan ekiplerin birbirine bağlanmasında köprü görevi görür.
 - üyeler. salatalık versiyon 4 kullanıyoruz
- Junit'i test aracı olarak kullanıyoruz. Junit, salatalık testlerini başlatmak ve aynı zamanda iddialarda bulunmak için kullanılır.
- Tarayıcıları otomatikleştirmek için Selenium WebDriver kullanıyoruz. selenium ile farklı tarayıcılarda testler yapabiliriz.
- Çerçevemiz, maven-cucumber-report eklentisini kullanarak ekran görüntüleriyle adım adım HTML raporları oluşturur.
- Çerçevemiz, veriye dayalı testleri destekler. Cucumber, senaryo ana hatlarını kullanarak veriye dayalı testleri yerel olarak destekler. The framework ayrıca apache POI kitaplıklarını kullanarak excel'den Veriye dayalı testler yapabilir.
- Çerçevem sayfa nesne modeline dayanıyor. sayfa nesnesi modeli, sayfadaki uygulamadaki sayfaları temsil ettiğimiz zamandır nesne sınıfları.
- bir web sürücüsü oluşturmak için fabrika modeli kullanıyoruz. webdriver sınıfımız, ne tür bir sürücüye bağlı olarak istemek. webdriver nesnesi için tekli desen kullanır
- URL, tarayıcı türü, oturum açma bilgileri vb. Gibi çerçevemizle ilgili önemli bilgileri depolamak için özellikler dosyasını kullanırız.
- Şirketimizde IntelliJ kullanıyoruz, ancak tutulmada da oldukça iyiyim.

Hangi uygulama katmanını test ediyor?

- Çerçevemizi kullanarak, uygulamanın kullanıcı arayüzünü, veritabanını ve API'sini test ediyoruz.
- **UI** → UI'yi test etmek için selenium web sürücüsü kullanıyoruz
- **Veritabanı** → veritabanını test etmek için JDBC kitaplıklarını kullanıyoruz
- **API** → **API**'yi test etmek için REST Assured kitaplıkları kullanıyoruz

Test türleri:

- **fonksiyonel , kabul testleri , duman testleri , regresyon testleri , entegrasyon testleri** yapıyoruz .
- Projemde sürüm kontrol aracı için git kullanıyoruz. (SVN)
- Otomatik duman ve regresyon testlerimizi planlamak ve çalıştırmak ve test sonuçlarını e-posta ile göndermek için Jenkins kullanıyoruz. Kullanma
 - jenkins, Xray eklentisini kullanarak testleri çalıştırabilir ve JIRA'yı test sonuçlarıyla güncelleyebiliriz.
- Çerçevemizde oturum açmak için Log4J kullanıyoruz.
- kod ve testlerin organizasyonu
- java kodumuzu düzenlemek için paketler kullanıyoruz, sayfa nesneleri, yardımcı programlar, StepDefinition, pojos, runners için paketlerimiz var
 - farklı yardımcı programlar kullanıyoruz, WebDriver için yardımcı programlarımız, tarayıcı yardımcı programları, excel yardımcı programı, yapılandırma yardımcı programı, Veritabanı yardımcı program, api ile ilgili yardımcı programlar, tarih yardımcı programı (takvimle ilgili uygulamayı test ettiğimiz için, tarihle ilgili çok şey yapıyoruz).
- özellikleri:

birçok özellik dosyamız var ve bunları düzenlemek için klasörler ve etiketler kullanıyoruz. her ana bileşen için klasörlerimiz var uygulama: rezervasyonlar, harita, hesap bilgileri ...

Jira'daki sorun numarasını özellik dosyasındaki etiketlerden biri olarak kullanırız, böylece onu Jira ile eşleştirebiliriz.

- özellikleri

rezervasyon
harita
hesap

- 134 -

Sayfa 135

Özellik: **Kullanıcı rolleri**

verilen içinde herhangi bir boş yer olup olmadığını öğrenmenin bir yolunu bulmanız gerekir

@login @ BRIT_3521 @smoke

Senaryo Özeti: <kullanıcı> türü olarak oturum açın

Kullanıcının açık taraf <kullanıcı> olarak oturum açtığı göz önüne alındığında

Ve zamanlama için uygun noktalar var

Ne zaman bir nokta için kullanıcı avı

Ardından kitap düğmesi <beklenen> görüntülenecek

Örnekler:

- | kullanıcı | beklenen |
- | takım üyesi | olmamalı |
- | takım lideri | gerekir |
- | öğretmen | gerekir |

Paralleleştirme → testlerimizi maven-surefire-eklentisini kullanarak paralel olarak çalıştırıyoruz. paralel testi destekleyen salatalık 4 kullanıyoruz doğal olarak.

Test verileri → test verilerimizi test / kaynaklar paketinde saklamak için özellik dosyalarını, excel dosyalarını kullanın

Yürütme akışı veya testleri nasıl çalıştırıyoruz

1. **cukes koşucusundan koşturmak**
 - cukes koşucusuna etiket koyarız ve sağ tıklayıp dosyayı çalıştırırız. cukes runner tüm eşleşen etiketleri çalıştırır.
2. **terminalden maven komutu olarak çalıştırın**
 - testleri başlatmak için mvn doğrulama veya mvn test komutunu giriyoruz. ve maven, pomda gösterilen cukes runner dosyalarını çalıştıracak

dosya. cukes runner dosyası, sahip olduğu etiketlerle eşleşen özellikleri çalıştırır.

GIT dallanma

git dallanmasını şimdi nasıl yapıyorsunuz?

Şu anda GRUP PROJELERİNDE NASIL YAPILIR:

Her ekip üyesi için ana şube ve ayrı şubeler vardır. birisi işi bitirdiğinde, kendi şubelerini inceledikten sonra ustalaşmak için birleştirilir.

NASIL YAPTIK?

projemde kişi için ustalık, geliştirme ve şubemiz vardı. bu nedenle, 2 otomasyon test cihazımız varsa,

usta

geliştirmek

test cihazı1

test cihazı2

her test kullanıcısı kendi şubelerine giriş yapar. incelendikten sonra şube geliştirmek için birleştirilir. ustayı birleştiriyoruz ve sadece bir sprint geliştirir.

🔗 görüşmede dallanma hakkında sorduklarında, otomasyon projesi dallanma stratejiniz hakkında konuşun.

Projenizde, kodunuz uygulama kod deposundan ayrı bir depodur. Otomasyon çerçevesi daha küçük bir kod tabanına sahiptir ve daha az insan dahil oldu. Böylece, daha az karmaşık dallanma politikasına sahip olabiliriz.

JAVA Mülakat Teknik Soruları

Ters Dize - Döngü İçin

```
public static String reverseString (String str) {  
    Dize ters = "";  
    for (int i = str.length () - 1; i >= 0; i--) {  
        ters += str.charAt (i);  
    }  
    tersine dönüş;  
}
```

EN İYİ YOL:

```
Dize adı = "Alper Aslan";  
String reversed = new StringBuilder (isim) .reverse (). ToString ();  
System.out.println (tersine çevrilmiş);
```

Ters Dize - StringBuilder

```
public static void main (String [] args) {  
    String str = "Merhaba Dünya";  
    StringBuilder sb = new StringBuilder (str);  
  
    Sistem. out .println (sb.reverse ()); }
```

Ters Dize - Karakter Dizisi

```
public void printReverse (karakter [] harfler, int boyut) {  
    for (int i = letters.length-1; i >= 0; i -) {  
        System.out.print ([i] harfleri);  
    }  
}
```

Asal sayı

```
public static boolean checkPrime (int n) {  
    eğer (n <= 1) {
```