unit test : Birim testi, kodun en küçük işlevsel birimini test ettiğiniz süreçtir.

Unit Testi methodlar ve fonksiyonlarımız için yazarız  
  
Unit Testin tooları

1. **LambdaTest**
2. **JUnit**
3. **NUnit**
4. **TestNG**
5. **PHPUnit**
6. **Mockito**
7. **Cantata**
8. **Karma**
9. **Mocha**
10. **TypeMock.**
11. **Jest**
12. **Jasmine**
13. **MsTest**

**En Çok kullanılan Toolar**

**1 Lambda Test:,**

LambdaTest, geliştiricileri ve test uzmanlarını 3000'den fazla gerçek tarayıcı, cihaz ve platform kombinasyonunda geniş ölçekte otomatik birim testi gerçekleştirmeleri için donatan yapay zeka destekli bir test düzenleme ve yürütme platformudur.

Özellikler

Gerçek masaüstü ve mobil ortamlarda popüler otomasyon test çerçeveleri ile otomatik birim testleri çalıştırın.

HyperExecute platformu ile sürüm döngülerini hızlandırın ve geliştiricilerden daha hızlı geri bildirim alın.

Yapay zeka destekli SmartUI platformu ile görsel birim testleri çalıştırın.

53'ten fazla ülkeden (ve şehirden) otomatik coğrafi konum testi gerçekleştirin.

CI/CD, proje yönetimi, kodsuz otomasyon ve daha fazlası için 120'den fazla üçüncü taraf entegrasyonuna erişin.

Yapay zeka destekli Test Analytics platformu ile birim test sonuçlarınız hakkında daha derin test içgörüleri elde edin.

https://www.lambdatest.com/

metin, yazılım, ekran görüntüsü, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturulduekran görüntüsü, metin, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

2. Junit Test

JUnit, Java'da tekrarlanabilir testler yazmak için basit bir çerçevedir. Birim test çerçeveleri için xUnit mimarisinin bir örneğidir.

Test yöntemlerini tanımlamak için @Test gibi ek açıklamaları kullanır. Ortak test verilerini kurmak ve yıkmak için @Before ve @After ek açıklamaları ile test fikstürlerini destekler

Junit'in Artıları:

Geniş benimsenme ve topluluk desteği.

Eclipse ve IntelliJ IDEA gibi IDE'lerle sorunsuz bir şekilde entegre olur.

Ek açıklamalar test kodunu temiz ve anlaşılması kolay hale getirir

Junit'in Eksileri:

Java ile sınırlıdır, diğer programlama dilleri için uygun değildir.

Yerel olarak gelişmiş mocking yeteneklerinden yoksundur.

Kullanım Durumu: Özellikle kurumsal ortamlarda güvenilir ve yaygın olarak desteklenen bir test çerçevesi arayan Java geliştiricileri için idealdir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

<https://semihsaydam.medium.com/unit-test-3-junit-da1a6497c83b> Bu linkten geniş kullanımını detaylı bir şekilde öğrenebilirsiniz.

3.NUnit (C#)

Unit, .NET dilleri için açık kaynaklı bir birim test çerçevesidir. JUnit'e benzer ancak .NET ortamı için tasarlanmıştır.

Veri odaklı testleri destekler ve testleri paralel olarak çalıştırabilir. Testleri ve test sınıflarını tanımlamak için [Test] ve [TestFixture] gibi nitelikler kullanır.

Nunit'in Artıları:

Güçlü parametrelendirilmiş test yetenekleri.

Paralel test yürütme için uygundur.

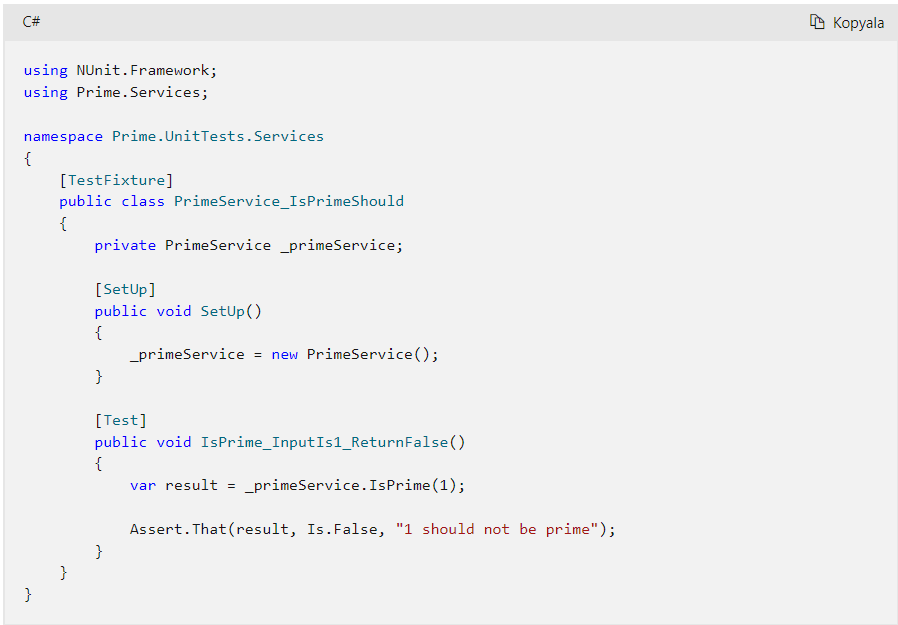
Güçlü bir assertion kütüphanesine sahiptir.

Nunit'in Eksileri:

MSTest'e kıyasla .NET Core ile daha az entegrasyon seçeneği.

Yeni başlayanlar için karmaşık olabilir.

Kullanım Durumu: Özellikle karmaşık uygulamalar için sağlam ve zengin özelliklere sahip bir çerçeveye ihtiyaç duyan .NET geliştiricileri için en uygunudur.

<

<https://learn.microsoft.com/tr-tr/dotnet/core/testing/unit-testing-with-nunit> Bu linkten Türkçe Dokümantasyonunu

<https://docs.nunit.org/> Bu Linkten de kendi dokümantasyonuna ulaşılabilir.

4. MSTest (C#)

MSTest, Microsoft'un Visual Studio IDE'ye entegre edilmiş resmi test çerçevesidir.

Yerleşik test koşucularına sahiptir ve testleri düzenlemek için test kategorilerini destekler. TestMethod] ve [TestClass] gibi öznitelikleri kullanır.

MSTEST'in Artıları (C#):

Visual Studio ve diğer Microsoft araçları ile sıkı entegrasyon.

Microsoft ekosistemine aşina olan geliştiriciler için kullanımı kolaydır

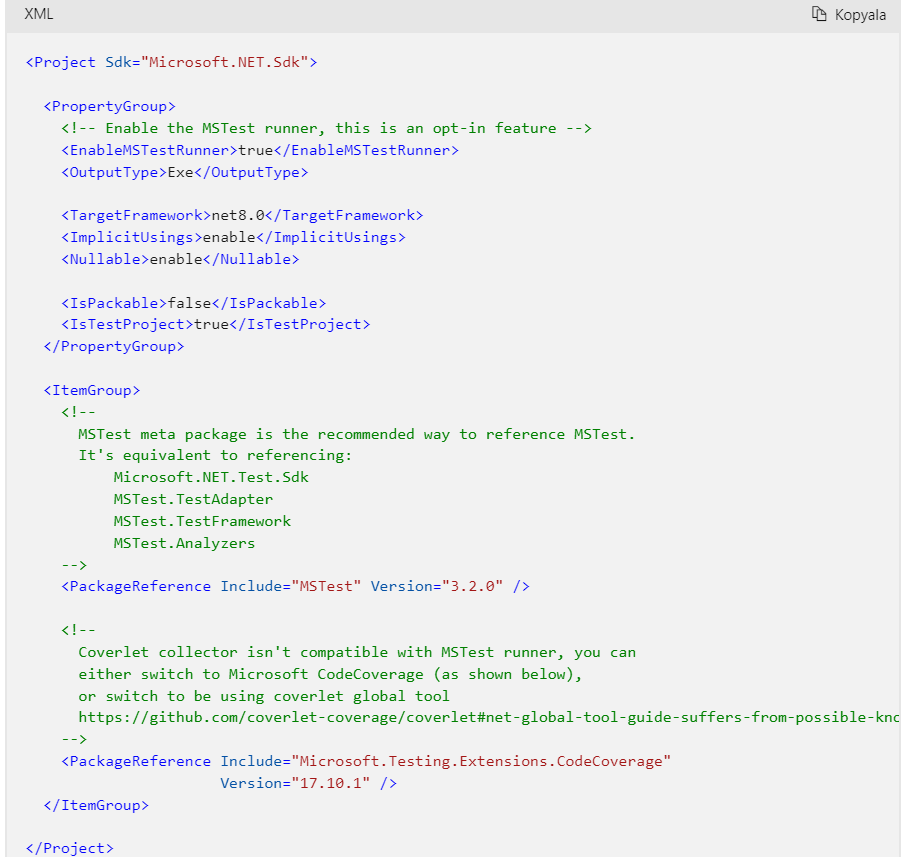
Veri odaklı testleri destekler.

MSTEST (C#)'ın Eksileri:

NUnit kadar zengin özelliklere sahip değildir.

Microsoft ekosistemi ile sınırlıdır.

Kullanım Örneği: Özellikle Visual Studio kullananlar olmak üzere Microsoft ekosistemine yoğun yatırım yapan ekipler için mükemmeldir.



<https://learn.microsoft.com/tr-tr/shows/on-net/writing-tests-with-mstest-v2> Ms Test ile ilgili Tutorial video

<https://learn.microsoft.com/tr-tr/dotnet/core/testing/unit-testing-mstest-runner-intro> Bu da Türkçe Kaynak

5. Mocha (JavaScript)

Mocha, Node.js üzerinde ve tarayıcıda çalışan esnek bir JavaScript test çerçevesidir ve asenkron testi basitleştirir

Davranış odaklı geliştirme (BDD), test odaklı geliştirme (TDD) ve diğer test stillerini destekler.

JavaScript için kullanılan yaygın programdır.

Mocha'nın Artıları (JavaScript):

Çok çeşitli eklentiler ve entegrasyonlar.

Zengin raporlama ve istisnaları test senaryolarına eşleme.

Kolay asenkron test desteği.

Mocha'nın (JavaScript) Eksileri:

Birlikte gelmediği için assertionlar için assertion kütüphaneleri (Chai gibi) gerektirir.

Büyük test süitlerinde yavaş olabilir.

Kullanım Durumu: Hem ön uç hem de arka uç testi için çok yönlü bir araca ihtiyaç duyan JavaScript geliştiricileri için uygundur.

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

<https://mochajs.org/> Mochanın kullanımı için genel Dökümantasyon

6. Jest (JavaScript)

Jest, sadeliğe odaklanan kolay bir JavaScript Test Çerçevesidir.

Projelerle çalışır: Babel, TypeScript, Node, React, Angular, Vue ve daha fazlası.

Avantajlar: Hızlı ve güvenlidir, kod kapsamı, kolay içe aktarma,great exeptions,

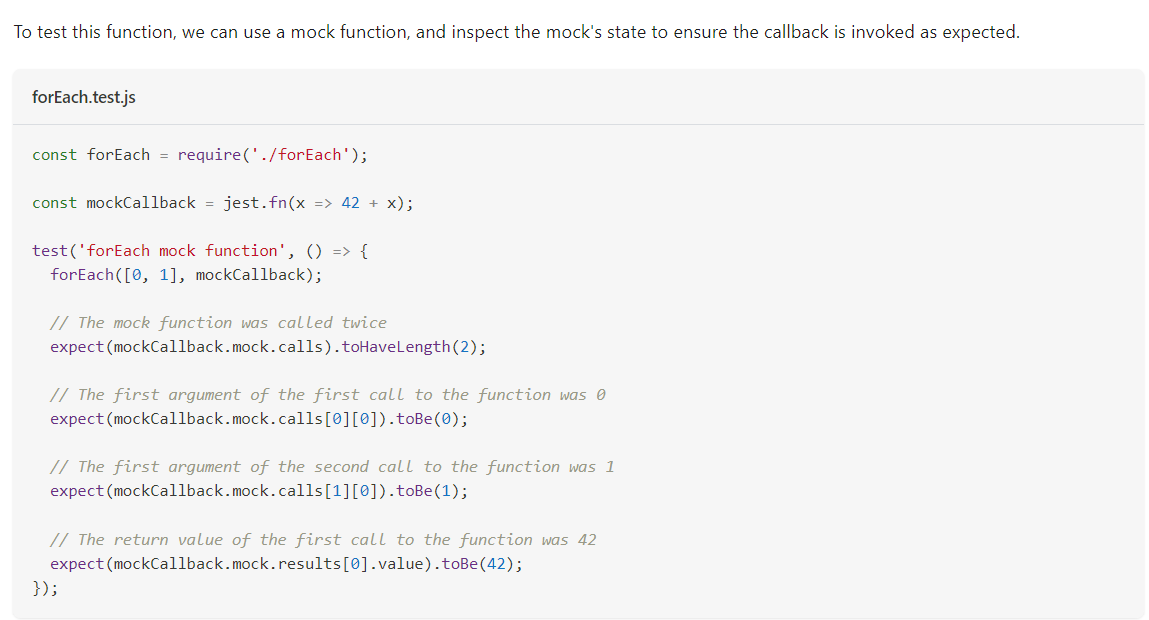
Eksiler: Çoğu durumda hızlı olsa da Jest, özellikle büyük projelerde diğer test çerçevelerinden daha yavaş olabilir, çok çeşitli özellikler ve eşleşmeler yeni başlayanları bunaltabilir.

JEST'I KİM KULLANIYOR?

Çok fazla insan! Geçen ay 300 milyondan fazla indirildi ve GitHub'da 11.000.000'dan fazla genel depoda kullanıldı.

Jest bu şirketlerde yaygın olarak kullanılmaktadır:

Facebook,X,NewYorkTimes,Spotify,Airbnb,Instagram



<https://jestjs.io/> Bu linkten dokümantasyonuna bakılarak kullanabilir.

7. PHPUnit(PHP)

PHPUnit, PHP için programcı odaklı bir test çerçevesidir. JUnit ve diğer test çerçevelerinden esinlenmiştir.

XUnit mimarisini takip eder. PHPUnit, test yöntemlerini tanımlamak için ek açıklamalar kullanır ve kurulum ve kaldırma yöntemlerini destekler.

PHPUnit'in (PHP) Artıları:

Yaygın olarak kullanılır ve iyi belgelenmiştir.

Veri odaklı testler için veri sağlayıcı yöntemlerini destekler.

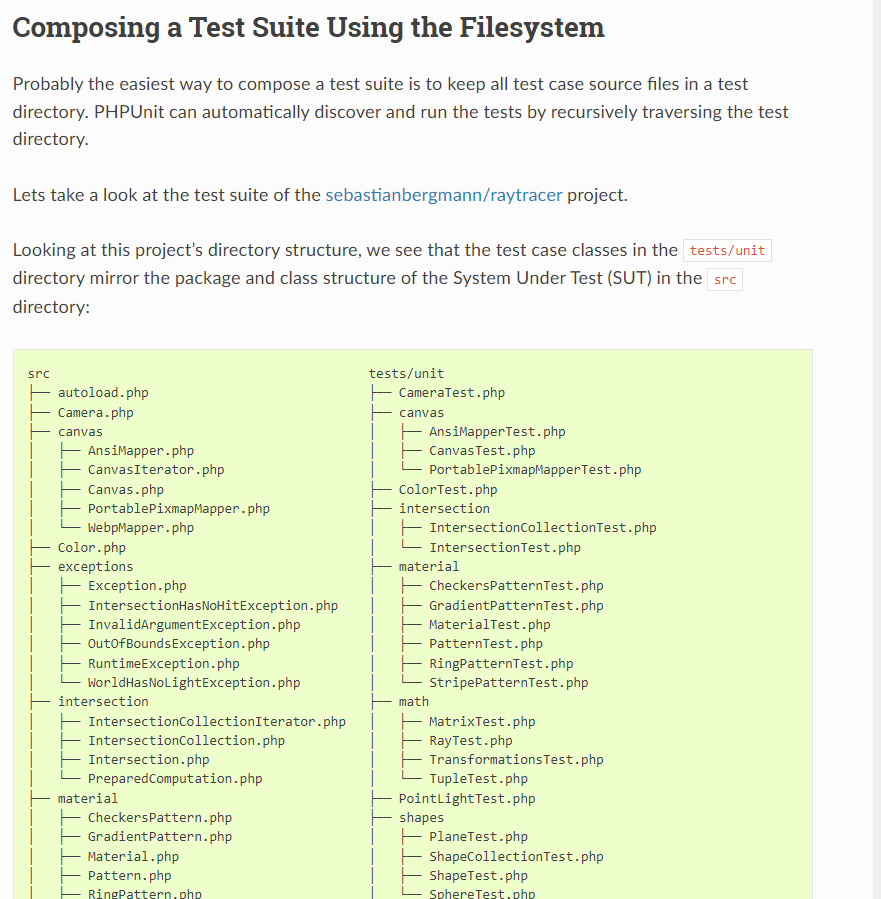
Birçok PHP projesi ve çerçevesi ile iyi entegrasyon.

PHPUnit'in (PHP) Eksileri:

Bazı PHP uygulamaları ile kurulumu zor olabilir.

Bağımsız mocking çerçevelerine kıyasla sınırlı mocking yetenekleri.

Kullanım Durumu: PHP geliştiricileri, özellikle Laravel ve Symfony gibi çerçevelerle çalışanlar için gereklidir.



<https://phpunit.de/documentation.html> Buradan daha iyi anlaşılabilir.

8. PyTest (Python)

PyTest, Python'un standart unittest modülüne boilerplate olmayan bir alternatiftir.

Testler için sınıf gerektirmez, parametrelendirilmiş testleri destekler ve kutudan çıkar çıkmaz unittest (deneme dahil) ve nose test paketlerini çalıştırabilir.

PyTest'in (Python) Artıları:

Basit sözdizimi, testleri yazmak ve okumak kolaydır.

Test durumunu yönetmek için güçlü fikstür sistemi.

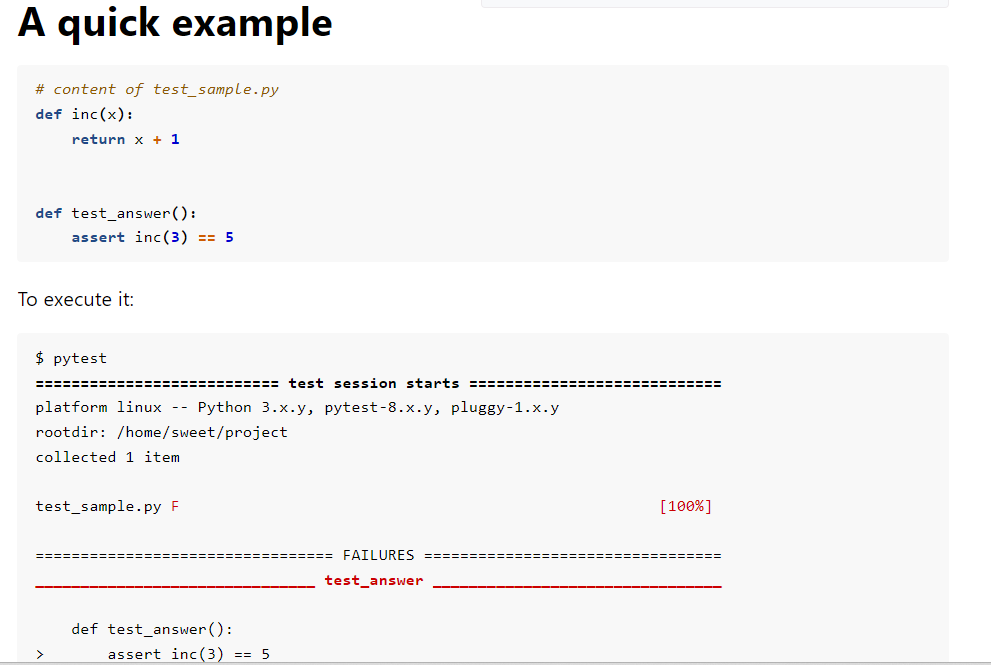
Kapsamlı eklenti sistemi.

PyTest'in (Python) Eksileri:

Gelişmiş özellikleri için öğrenme eğrisi.

Bazen büyük test paketleri için unittest'ten daha yavaş.

Kullanım Örneği: Her seviyedeki Python geliştiricileri için harika, çok çeşitli test ihtiyaçları için güçlü ancak kullanıcı dostu özellikler sunuyor.



<https://docs.pytest.org/en/8.2.x/> Bu Linkten Dokümantasyonuna ulaşılabilir.

9. Mockito

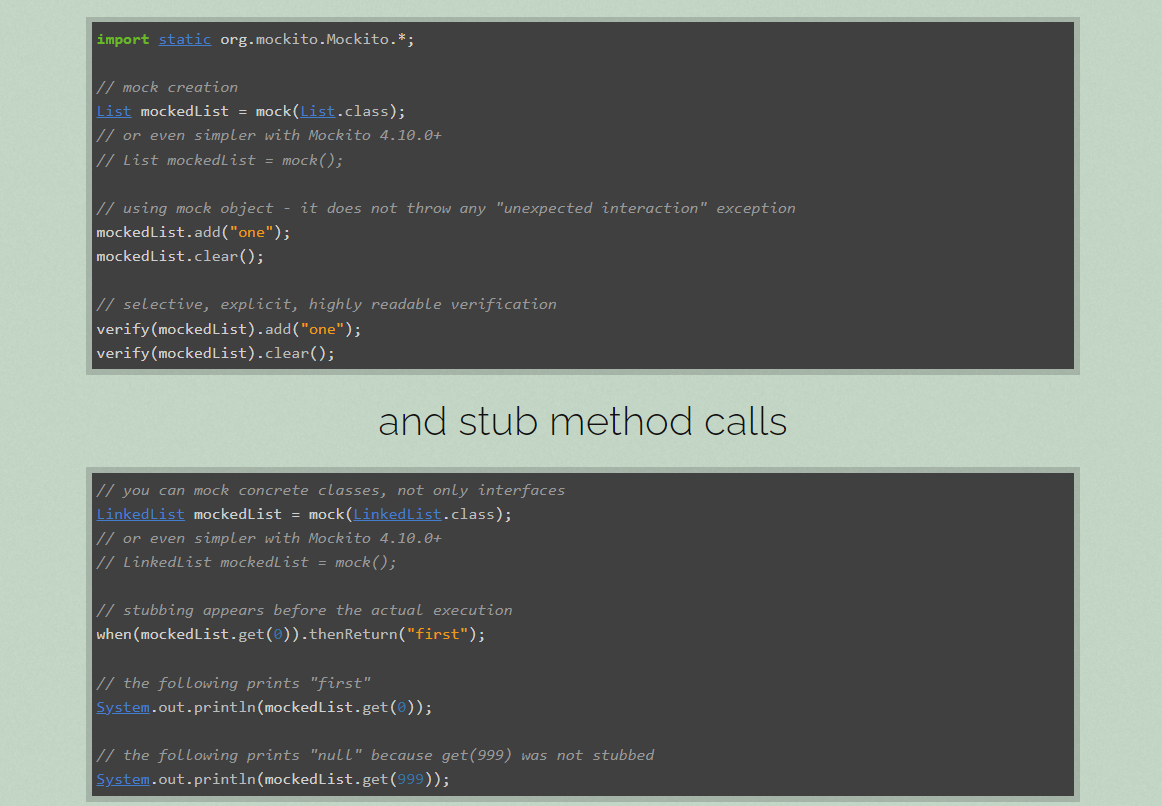
Mockito, Java için geliştirilmiş, yaygın olarak kullanılan bir mocking framework'üdür. Birim testlerinde, gerçek bağımlılıklar yerine sahte (mock) nesneler kullanarak test yapmayı kolaylaştırır. Bu sayede, testlerin daha izole ve hızlı olması sağlanır.

**Avantajları**

1. **Kolay Kullanım**: Mockito'nun API'si oldukça sezgiseldir ve öğrenmesi kolaydır. Kod okunabilirliğini artırır.
2. **Esneklik**: Hem davranış odaklı (behavior-driven) hem de etki odaklı (state-driven) testleri destekler.
3. **Geniş Topluluk ve Destek**: Geniş bir kullanıcı kitlesine sahiptir ve birçok dokümantasyon, rehber ve topluluk desteği mevcuttur.
4. **Daha İyi Test İzolasyonu**: Gerçek bağımlılıkları kullanmak yerine mock nesneler kullanarak testlerinizi izole edebilirsiniz. Bu, testlerin hızlı ve tutarlı olmasını sağlar.
5. **Kolay Entegrasyon**: JUnit ve TestNG gibi popüler test framework'leri ile kolayca entegre edilebilir.
6. **Flexible Verification**: Çağrıların sayısını ve sırasını doğrulama gibi esnek doğrulama seçenekleri sunar.

**Dezavantajları**

1. **Gerçeklikten Uzak Testler**: Mock nesneler kullanıldığında, testler gerçek dünyadaki senaryolara her zaman birebir uyum göstermeyebilir.
2. **Bakım Zorluğu**: Mock nesneler ve onların davranışları zamanla değişebilir, bu da testlerin bakımını zorlaştırabilir.
3. **Aşırı Kullanım Riski**: Mocking'in aşırı kullanılması, testlerin gereksiz derecede karmaşıklaşmasına ve zor anlaşılmasına neden olabilir.
4. **Yetersiz Kapsam**: Mock nesneler, yalnızca belirli davranışları test eder. Bağımlılıkların gerçek uygulamaları üzerindeki etkilerini gözden kaçırabilir.
5. **Test Yanlılığı**: Mock nesneler, belirli bir davranış sergilemek üzere programlandıkları için testler bazen yanlı olabilir ve beklenmedik durumları göz ardı edebilir.



<https://site.mockito.org/> Bu linkten dokümana ulaşılabilir

2. Entegrasyon Testi (Integration Testing)

Top 10

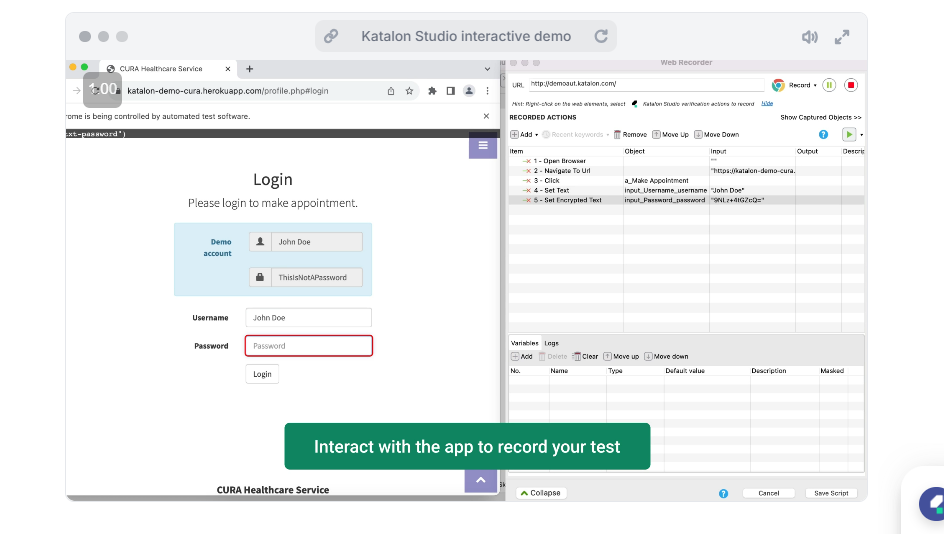
1. Katalon
2. Postman
3. SoapUI
4. Citrus
5. Selenium
6. IBM Rational Integration Tester
7. Leapwork
8. Tessy
9. LDRA
10. TestSigma
11. Katalon

Katalon Platform, web, mobil, masaüstü ve özellikle entegrasyon testinin çok önemli bir parçası olan API testi dahil olmak üzere her tür AUT için hepsi bir arada bir entegrasyon test aracıdır.

Katalon'da entegrasyon testi, sağladığı kapsamlı özellikler sayesinde özellikle etkilidir. Yazılım bileşenleri arasındaki ve hatta bu bileşenler ile harici hizmetler arasındaki etkileşimleri kontrol etmek için hızlı bir şekilde test senaryoları oluşturabilirsiniz (örneğin, bir finans yazılımının tüccarlar için finansal haberleri almak üzere bir web hizmetine etkili bir şekilde bağlanıp bağlanmadığını test etmek gibi). Katalon'un hepsi bir arada yapısı, AUT'ler arasında test senaryoları oluşturmanıza olanak tanır.

Sadece bu da değil, Katalon Studio'da herhangi bir kodlama yapmadan test komut dosyaları oluşturabilirsiniz. Yerleşik anahtar kelimeleri (gerçekleştirmek istediğiniz eylem için temelde kod parçacıklarıdır) sürükleyip bırakmanız veya test adımları dizinizi kaydetmek için Kaydet ve Oynat özelliğini kullanmanız yeterlidir. Katalon tüm bunları farklı ortamlarda çalıştırabileceğiniz yeniden çalıştırılabilir test komut dosyalarına dönüştürür.

Ayrıca test senaryolarınızı ve test nesnelerinizi Nesne Deposunda net bir hiyerarşi ile kolayca yönetebilirsiniz. Test yürütme söz konusu olduğunda, Katalon size çok çeşitli tarayıcılara ve mobil sürümlere (hem iOS hem de Android) erişim sağlar ve yerel, uzaktan veya bulut üzerinde çalıştırmayı seçebilirsiniz. Test çalıştırmalarından sonra Katalon, bilinçli kararlar almanız için ayrıntılı test raporları oluşturur.



<https://katalon.com/katalon-platform> websitesine buradan ulaşılabilir.

1. Postman

Postman, API'lerin oluşturulması, test edilmesi ve belgelenmesi sürecini basitleştirmek için tasarlanmış kapsamlı bir API geliştirme ve test platformudur. Ana kullanım alanı API testi olsa da, Postman entegrasyon testi için de yaygın olarak kullanılmaktadır. Harici dijital hizmetlerle entegrasyon senaryolarını test etmek için API uç noktalarını kapsayan test paketlerini kolayca oluşturabilirsiniz.

Öne Çıkan Özellikler:

API testi oluşturma, yönetme, yürütme ve raporlamaya kadar tüm API testi yaşam döngüsünü destekleyin

Öncelikli olarak RESTful API'ler için tasarlanmıştır, ancak SOAP isteklerini de belirli bir ölçüde işleyebilir.

İstek koleksiyonları oluşturarak ve JavaScript kullanarak test komut dosyaları tanımlayarak otomatik entegrasyon testini etkinleştirin. Test komut dosyaları, kolay özelleştirme için doğrudan Postman arayüzünde yazılır.

Henüz mevcut olmamaları durumunda arka uç hizmetlerinin davranışını simüle etmek için API mocking oluşturulmasına izin verin.

Veri odaklı test desteklenir

Mevcut CI/CD işlem hattınızla sorunsuz bir şekilde entegre edin

Entegrasyon testleriniz için farklı konfigürasyonları ve uç noktaları yönetmenize yardımcı olacak bir ortam yöneticisi mevcuttur.

İşbirliği özellikleri dahil

<https://www.postman.com/> Websitesi

metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

3. SoapUI

SoapUI, SOAP ve REST API'leri dahil olmak üzere web hizmetlerini test etmek için kullanılan popüler bir açık kaynak aracıdır. Ayrıca, kullanıcıların test senaryoları oluşturmasına, test adımlarını tanımlamasına ve test verilerini kolayca yapılandırmasına olanak tanıyan kullanıcı dostu bir grafik arayüze sahip bir entegrasyon test aracı olarak da yaygın olarak kullanılmaktadır.

Öne Çıkan Özellikler:

SoapUI, hem SOAP (Basit Nesne Erişim Protokolü) hem de REST (Temsili Durum Aktarımı) web servislerini destekleyerek çeşitli API'lerle uyumlu hale getirir.

API yanıtlarını doğrulamak için durum kodları, içerik, başlıklar ve daha fazlası dahil olmak üzere çok çeşitli onay türleri sunar

Özel mantık oluşturmak ve karmaşık test senaryolarını otomatikleştirmek için komut dosyası oluşturma modu.

Kullanıcılar, giriş verilerini kolayca değiştirmek için test senaryolarını parametrelendirebilir, böylece test senaryosu yapısını değiştirmeden farklı senaryoları test etmeyi kolaylaştırır.

Güvenlik testi ve yük testi de desteklenir

Bmw Apple Microsoft Vodofone gibi şirketler tarafından kullanılmaktadır.

https://www.soapui.org/

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

4 Citrus

Citrus, mesajlaşma sistemleri, API'ler, veritabanları ve çok daha fazlasının otomatik entegrasyon testi için tasarlanmış açık kaynaklı Java tabanlı bir entegrasyon testi çerçevesidir. Bir kurumsal mimaride farklı bileşenler ve sistemler arasındaki etkileşimleri içeren senaryoları test etmek için özel olarak tasarlanmıştır.

Citrus, entegrasyon testleri sırasında hem istemci hem de tüketici olarak hareket ederek ikili bir rol oynayabilir. Bir istemci olarak Citrus, Test Edilen Sisteme mesajlar (istekler) gönderir. Tüketici olarak, Test Edilen Sistemden mesajlar (yanıtlar) alır ve işler, böylece sistemler arasındaki gerçek dünya etkileşimlerini simüle eder.

Öne Çıkan Özellikler:

Citrus, kendisi ve Test Edilen Sistem arasında değiş tokuş edilen mesajları doğrulayabilir, mesaj içeriğini, başlıkları, ekleri ve diğer kontrol verilerini kontrol ederek entegrasyon testi gereksinimlerinin beklenen kriterlerini karşıladığından emin olabilir.

Citrus, test uzmanlarının test içinde beklenen davranışı ve iddiaları tanımlamasını kolaylaştırmak için Java akıcı bir API sağlar.

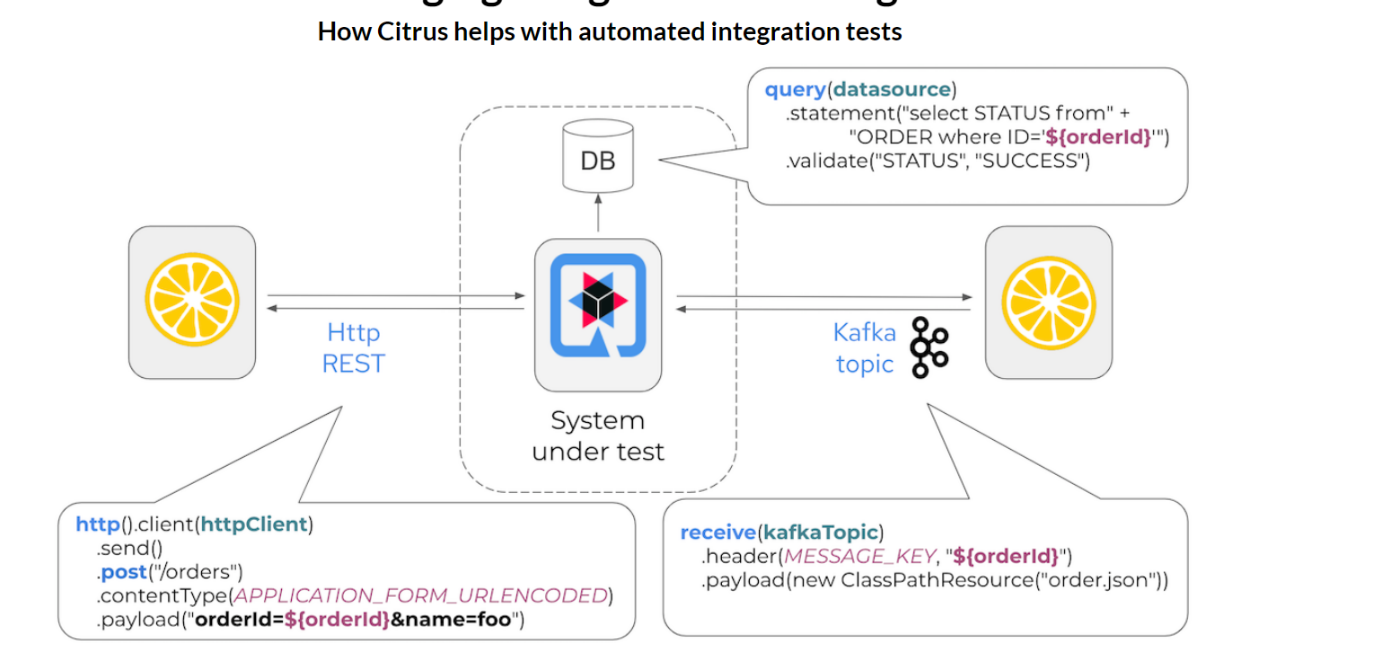
CI/CD boru hattında otomatik test yürütme

Microservice uygulamalarının otomatik entegrasyon testini desteklemek için Apache Camel, Arquillian, Kubernetes ve Docker gibi çerçevelerle entegrasyon

Sağlam, kurumsal düzeyde entegrasyon testleri oluşturmak için Spring framework ile entegre edin.

BDD testi desteklenir (hem teknik hem de teknik olmayanların anlayabileceği doğal dilde testler yazma)

Açık kaynak kodlu olması biraz avantaj sağlar.



<https://citrusframework.org/> buradan ulaşılabilir.

5.Selenium

Selenium, web tarayıcısı etkileşimlerini otomatikleştirme yeteneği nedeniyle geliştiriciler tarafından yaygın olarak tercih edilen popüler bir açık kaynaklı otomasyon test çerçevesidir. Sadece entegrasyon testi değil, Selenium çok çeşitli web senaryolarını test etmek için de kullanılabilir. Düğmelere tıklama, form doldurma, sayfalar arasında gezinme ve çok daha fazlası gibi eylemleri kodlayabilir ve otomatikleştirebilirsiniz.

Bununla birlikte, Selenium'dan tam olarak yararlanmak için test uzmanının iyi düzeyde kodlama deneyimine sahip olması gerekir. Bu yalnızca bir çerçevedir ve test senaryolarını gerçekten otomatikleştirmek için test uzmanlarının oluşturdukları test senaryolarını aktif olarak yazmaları ve sürdürmeleri gerekir. Veriye dayalı test veya raporlama özellikleri için yerleşik çerçeveler yoktur, bu nedenle QA ekiplerinin de bunları kendi başlarına oluşturmaları gerekir. Bu yüksek öğrenme eğrisi, ekibe sağladığı özelleştirme düzeyi düşünüldüğünde buna değer.

Öne Çıkan Özellikler:

Birden fazla web tarayıcısını destekleme (Chrome, Firefox, Edge, Safari, vb.)

Tercih ettiğiniz dilde komut dosyaları yazma (Java, Python, C#, Ruby ve JavaScript()

Testleri görünür bir UI (headless browser) olmadan çalıştırabilirsiniz, ancak UI testi de desteklenir ve size kapsamlı bir entegrasyon testi deneyimi sunar

Selenium Grid ile testleri paralel olarak yürütme

Senkronizasyon sorunlarını ele almak için açık ve örtük bekleme mekanizmaları

Selenium, test yürütme sırasında ekran görüntüleri ve videolar yakalamanıza olanak tanıyarak hata ayıklama ve raporlamaya yardımcı olur.

Geniş ve aktif kullanıcı topluluğu vardır

Açık kaynak kodlu olduğundan avantajlıdır.

Popülerdir.



<https://www.selenium.dev/documentation/test_practices/overview/> buradan ulaşın

1. LeapWork

Leapwork, test otomasyonu, süreç otomasyonu ve robotik süreç otomasyonu (RPA) için tasarlanmış kodsuz bir otomasyon platformudur. Kullanıcıların kodlama veya komut dosyası yazmaya gerek kalmadan otomasyon iş akışları ve test senaryoları oluşturmalarını sağlar.

Bir entegrasyon test aracı olarak Leapwork'ün kodsuz yaklaşımı, hem test uzmanlarının hem de teknik olmayan kullanıcıların görsel bir arayüz aracılığıyla entegrasyon test senaryoları ve iş akışları tasarlamasına olanak tanıyarak çok çeşitli ekip üyeleri için erişilebilir olmasını sağlar.

Öne Çıkan Özellikler:

Tek bir satır kod yazmak zorunda kalmadan entegrasyon test senaryoları oluşturmak için kullanımı kolay yapı taşları

Veri odaklı otomasyon kutudan çıkar çıkmaz dahil.

Herhangi bir web öğesini yakalamak için işaretleyin ve tıklayın. Bozuk test komut dosyalarını düzeltmek için desteklenen kendi kendini iyileştirme mekanizması.

Düşük öğrenme eğrisi

SAP, Windows, Citrix, Java, IBM, Oracle ve diğer birçok uygulama üzerinde çalışarak kapsamlı entegrasyon testlerine olanak tanır

Kutudan çıkar çıkmaz programlama ve geçici çalıştırma.

DevOps'un diğer üçüncü taraf sistemlerle entegre olmasına yardımcı olmak için yerleşik REST API vardır.

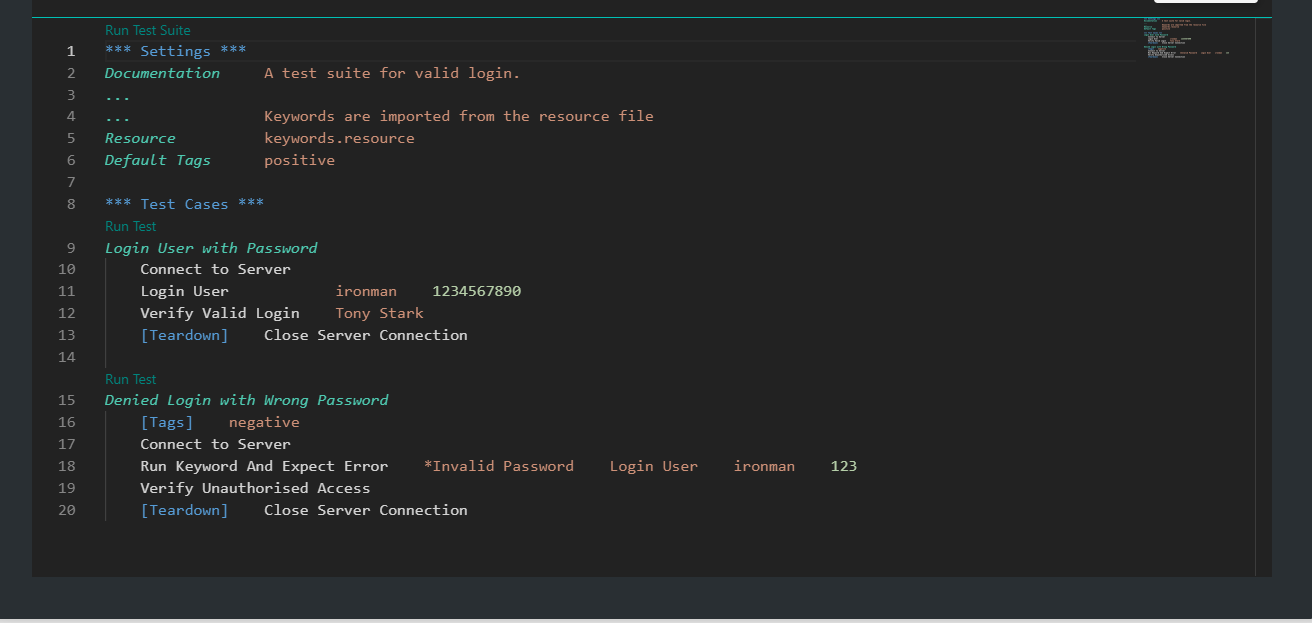
https://www.leapwork.com/test-automation

1. Sistem Testi
2. RobotFrameWork

Robot Framework, test otomasyonu ve robotik süreç otomasyonu (RPA) için açık kaynaklı bir otomasyon çerçevesidir. Robot Framework Foundation tarafından desteklenir ve sektörde yaygın olarak kullanılır.

İnsan dostu ve çok yönlü sözdizimi anahtar kelimeler kullanır ve Python, Java ve diğer dillerdeki kütüphaneler aracılığıyla genişletmeyi destekler.

Lisans ücreti olmadan kapsamlı otomasyon için diğer araçlarla entegre olur ve yüzlerce 3. parti kütüphaneye sahip zengin bir topluluk tarafından desteklenir.



<https://robotframework.org/>

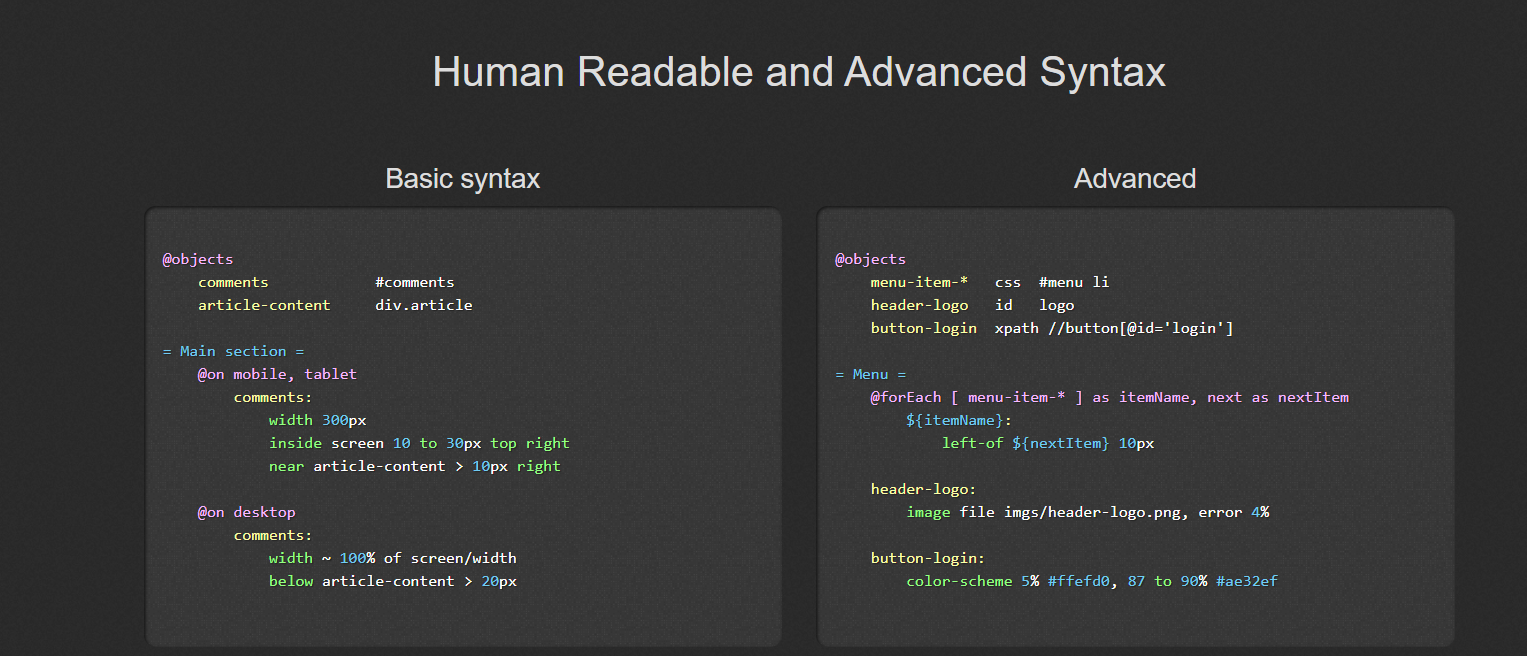
2.JMeter

Performans testlerinizi çalıştırmanıza ve otomatikleştirmenize yardımcı olacak birçok kullanışlı ve zaman kazandıran özellik kazanırken, herhangi bir değişiklik yapmak zorunda kalmadan JMeter testlerinizi Testable platformunda çalıştırın.Hem Basit Hem de Karmaşık Test Senaryoları Getirmek KolaySonsuz Ölçeklenebilir Dağıtılmış JMeter MotorlarıDünya Çapında Otomatik JMeter Motor Sağlama

<https://testable.io/>

3. Gallen FrameWork

Düzen testi her zaman karmaşık bir görev gibi görünüyordu. Galen Framework basit bir çözüm sunar: nesnelerin sayfa üzerinde birbirlerine göre konumlarını test edin. Özel bir sözdizimi ve kapsamlı kurallar kullanarak hayal edebileceğiniz herhangi bir düzeni tanımlayabilirsiniz



<https://galenframework.com/>

1. Selenium

Selenium, web uygulamalarını otomatikleştirmek ve test etmek için kullanılan açık kaynaklı bir araçtır. Farklı tarayıcılar üzerinde testler gerçekleştirmenizi sağlar ve Java, C#, Python, Ruby gibi birçok programlama dili ile entegre olabilir. Selenium'un sunduğu çeşitli bileşenler sayesinde, web uygulamalarının kullanıcı etkileşimlerini ve fonksiyonlarını detaylı bir şekilde test edebilirsiniz.

Yukarıda açıklamıştım unit testte.

**Avantajlar**

1. **Çapraz Tarayıcı Desteği**: Selenium, Chrome, Firefox, Internet Explorer, Edge, Safari gibi çeşitli tarayıcılarda çalışabilir, bu da uygulamanızın farklı tarayıcılarda nasıl performans gösterdiğini test etmenize olanak tanır.
2. **Çoklu Dil Desteği**: Java, C#, Python, Ruby gibi birçok programlama dili ile entegre olabilir, bu da mevcut teknoloji yığınınıza uygun bir dil seçmenizi sağlar.
3. **Açık Kaynak**: Ücretsizdir ve geniş bir kullanıcı topluluğuna sahiptir, bu da sürekli güncellemeler ve topluluk desteği anlamına gelir.
4. **Genişletilebilirlik**: Diğer test araçları ve framework'lerle (JUnit, TestNG, Cucumber) entegre edilebilir, bu da test süreçlerinizi genişletmenize ve optimize etmenize olanak tanır.
5. **Paralel Test Çalıştırma**: Selenium Grid kullanarak testlerinizi paralel olarak farklı tarayıcı ve platformlarda çalıştırabilirsiniz, bu da test süresini önemli ölçüde azaltır.
6. **Gerçekçi Testler**: WebDriver, gerçek tarayıcı oturumlarında çalışır, bu da testlerin gerçek kullanıcı etkileşimlerine yakın olmasını sağlar.

**Dezavantajlar**

1. **Bakım Zorluğu**: Web uygulamaları sürekli olarak değiştiği için, Selenium testlerini güncel tutmak zor olabilir. DOM yapısındaki değişiklikler testlerin kırılmasına neden olabilir.
2. **Performans**: Testlerin çalıştırılması zaman alabilir, özellikle büyük bir test setine sahipseniz. Paralel çalıştırma bu sorunu hafifletebilir, ancak her zaman yeterli olmayabilir.
3. **Dinamik İçerik**: AJAX ve dinamik içeriklerle çalışan uygulamalarda testlerin zamanlamasını ayarlamak zor olabilir. Bu tür testlerde doğru beklemeleri ayarlamak kritik önem taşır.
4. **Başlangıç Maliyeti**: Selenium'u öğrenmek ve verimli bir şekilde kullanmak zaman ve çaba gerektirir. Kullanıcıların Selenium API'sine ve test otomasyonu konseptlerine hakim olması gerekir.
5. **Görsel Testler**: Selenium, görsel doğrulama için tasarlanmamıştır. Görsel testler yapmak için ek araçlar (örneğin, Applitools) kullanmak gerekebilir.
6. **Mobil Test Desteği**: Selenium, mobil testler için doğrudan destek sağlamaz. Mobil testler için Appium gibi ek araçlar kullanmak gereklidir.

<https://www.selenium.dev/>

4.Kabul Testi

1. Zephyr

Zephyr, kullanıcı kabul testleri için destek içeren kapsamlı bir test yönetim aracıdır. Sezgisel arayüzü ve gelişmiş özellikleri ile Zephyr, tüm yazılım geliştirme yaşam döngüsü boyunca test faaliyetlerini yönetmek için popüler bir seçimdir.

Zephyr'in Özellikleri

Test senaryosu yönetimi: Zephyr, kullanıcı kabul testleri için test senaryoları oluşturmanıza ve yönetmenize olanak tanıyarak testlerin düzenlenmesini ve yürütülmesini kolaylaştırır.

Test yürütme: Test sonuçlarını takip etme ve otomatik olarak rapor oluşturma yeteneği de dahil olmak üzere kullanıcı kabul testlerini yürütmek için güçlü bir platform sağlar.

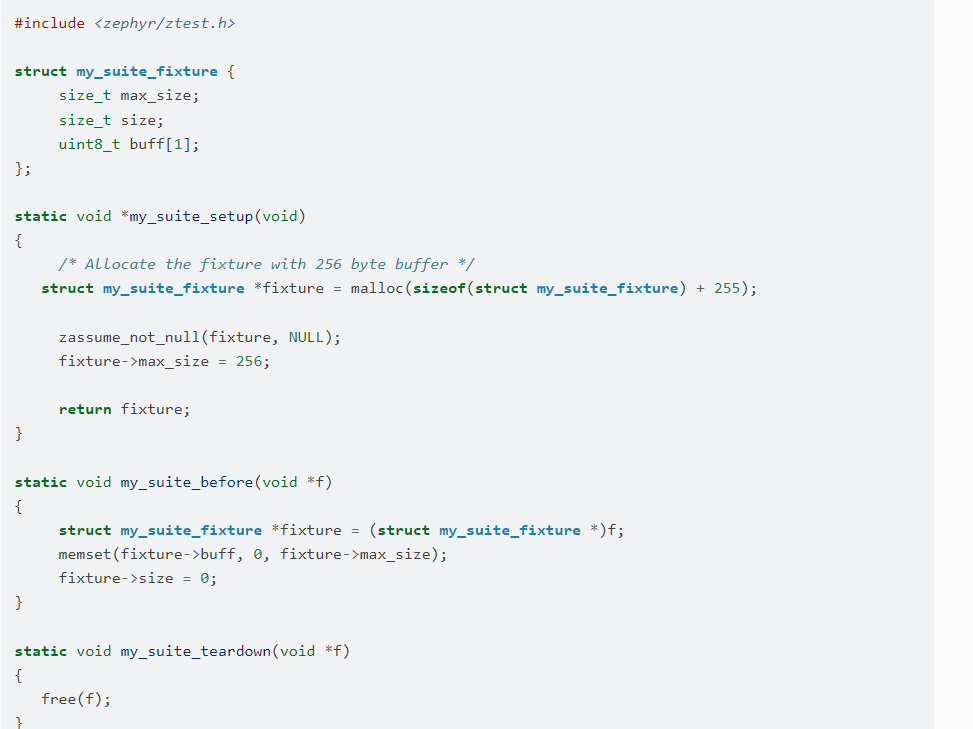
Entegrasyonlar: Zephyr, JIRA, Confluence ve Bamboo dahil olmak üzere çeşitli diğer test araçları ve platformlarıyla entegre olarak test faaliyetlerini genel yazılım geliştirme iş akışınıza entegre etmeyi kolaylaştırır.Gerçek zamanlı işbirliği: Birden fazla kullanıcının aynı test vakası üzerinde aynı anda çalışmasına olanak tanıyan ve tüm ekip üyelerine gerçek zamanlı güncellemeler sağlayan güçlü işbirliği özellikleri içerir.Zephyr'in Sınırlamaları :

Zephyr'in fiyatlandırması, küçük ekipler veya sınırlı bütçeye sahip kuruluşlar için sınırlayıcı bir faktör olabilir.

Bazı kullanıcılar, özellikle belirli özelliklerin ve iş akışlarının karmaşıklığı konusunda platformun kullanıcı arayüzü ile ilgili sorunlar bildirmiştir.

Genel olarak Zephyr, test yönetimi ve yürütme için gelişmiş özellikler sağlayan, kullanıcı kabul testi için güçlü ve kapsamlı bir araçtır. Zephyr hakkında daha fazla bilgi edinmek ve kendi test ihtiyaçlarınız için kullanmaya başlamak için web sitesini ziyaret edin.

<https://docs.zephyrproject.org/latest/develop/test/ztest.html>



1. TestRail

 **Açıklama**: TestRail, test yönetimi için kullanılan bir araçtır. Test senaryolarının planlanması, yürütülmesi ve raporlanması için kapsamlı bir çözüm sunar.

 **Avantajları**:

* Kullanıcı dostu arayüz ve kolay test yönetimi.
* Gelişmiş raporlama ve izlenebilirlik özellikleri.
* Farklı test otomasyon araçları ile entegrasyon imkanı.

 **Dezavantajları**:

* Ücretli bir araçtır.
* Büyük projelerde lisans maliyetleri artabilir.



<https://www.testrail.com/>

3.UserTesting

Doğrudan, gerçek zamanlı kullanıcı geri bildirimi

Hızlı ve kolay sayfa içi ve uygulama içi geri bildirim formları

Sorun noktalarını, hayal kırıklıklarını ve memnun olmadıkları UX karşılaşmalarını daha iyi anlamak için kullanıcıların uygulamanızla yaşadıkları ilk deneyimden içgörüler elde edin.

Şunun için en iyisi: UX ve UAT araçlarının tamamına ihtiyaç duyan büyük kuruluşlar.

Artıları Kurumsal düzeyde bir UX ve UAT araçları ve özellikleri paketi.

Eksileri: Geniş özellik ve araç yelpazesi. Daha küçük müşteri projeleri üzerinde çalışan daha küçük ekipler ve web geliştirme ajansları için çok fazla olabilir.

Alternatifler: Kurumsal sınıf çözümler için yok.

Fiyatlandırma: Kullanıma göre kişiye özel planlar.

https://www.usertesting.com/