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.

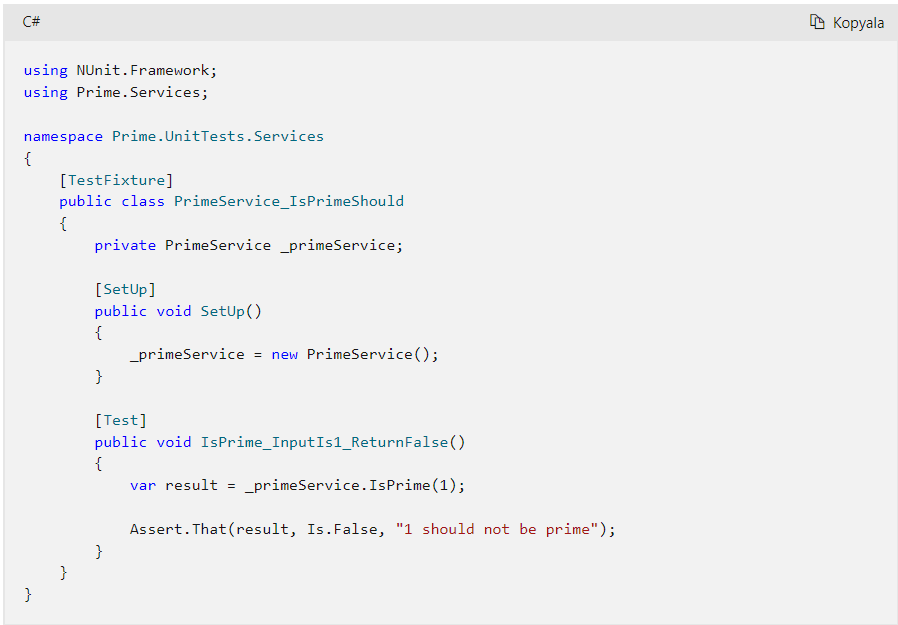
Güçlü bir assertion kütüphanesine sahiptir.

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,

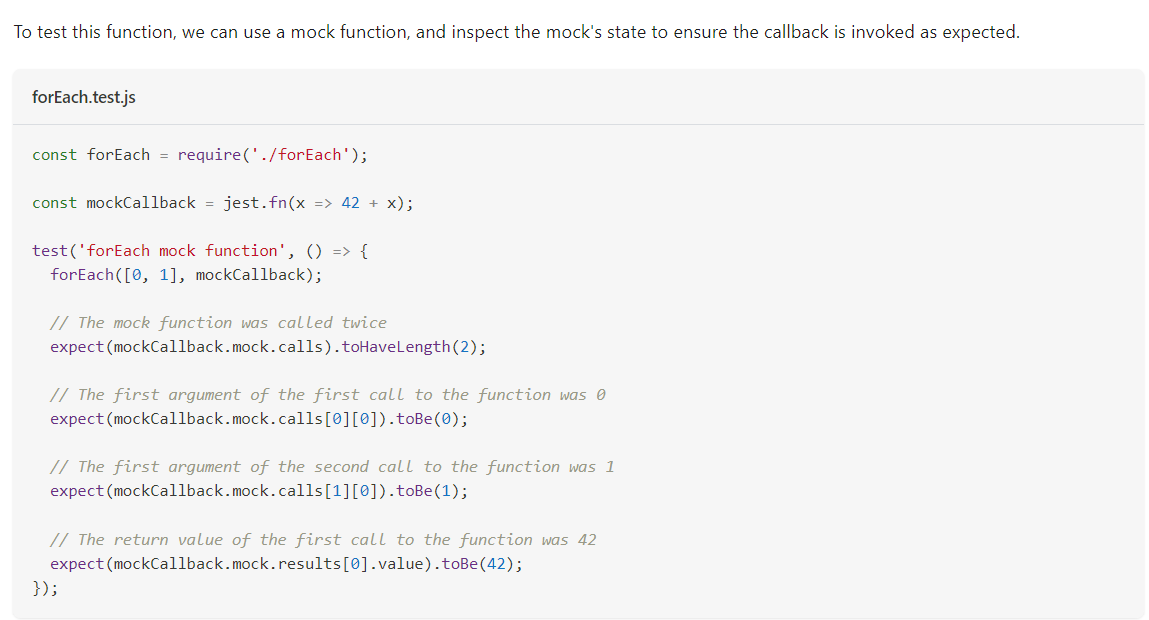
Eksiler: Çoğu durumda hızlı olsa da Jest, özellikle büyük projelerde diğer test çerçevelerinden daha yavaş olabilir, çok çeşitli özellikler ve eşleşmeler yeni başlayanları bunaltabilir.

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.

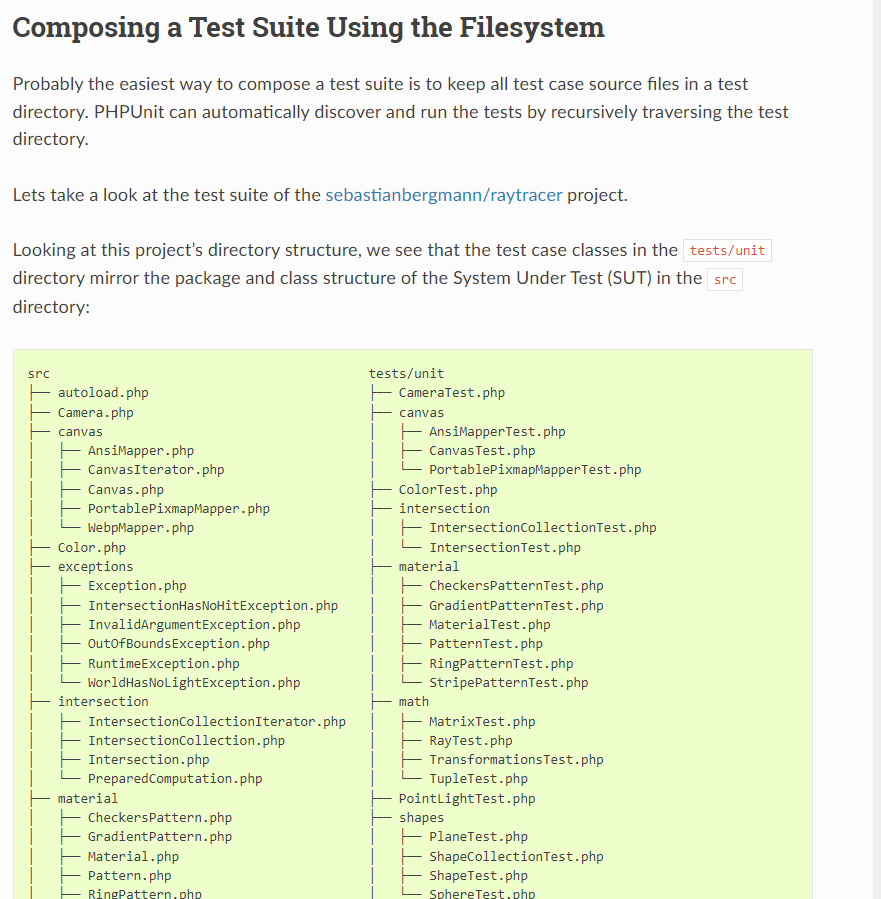
Birçok PHP projesi ve çerçevesi ile iyi entegrasyon.

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.

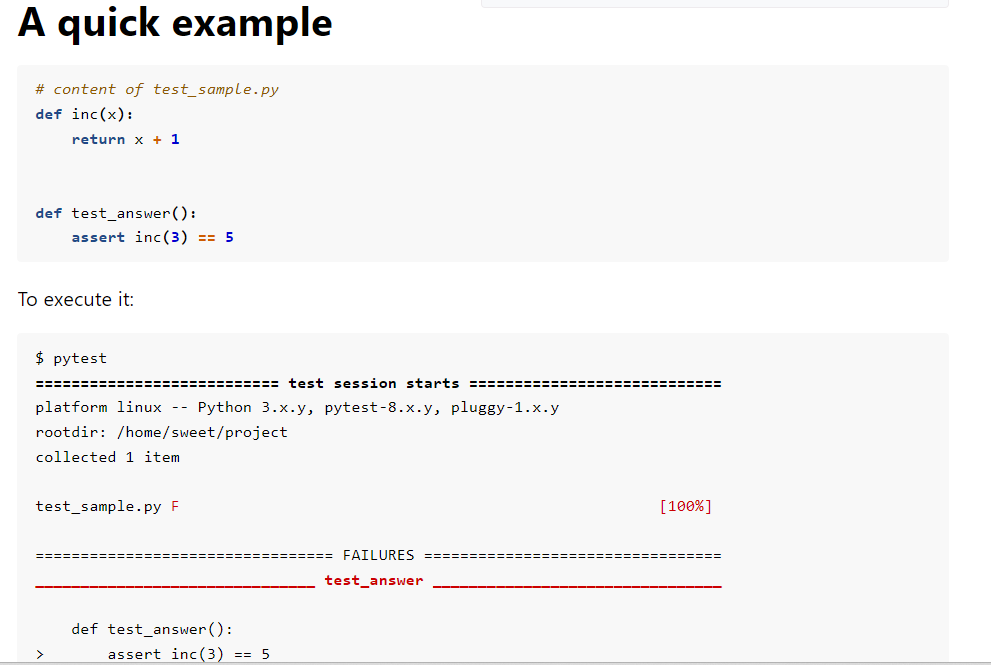
Kapsamlı eklenti sistemi.

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

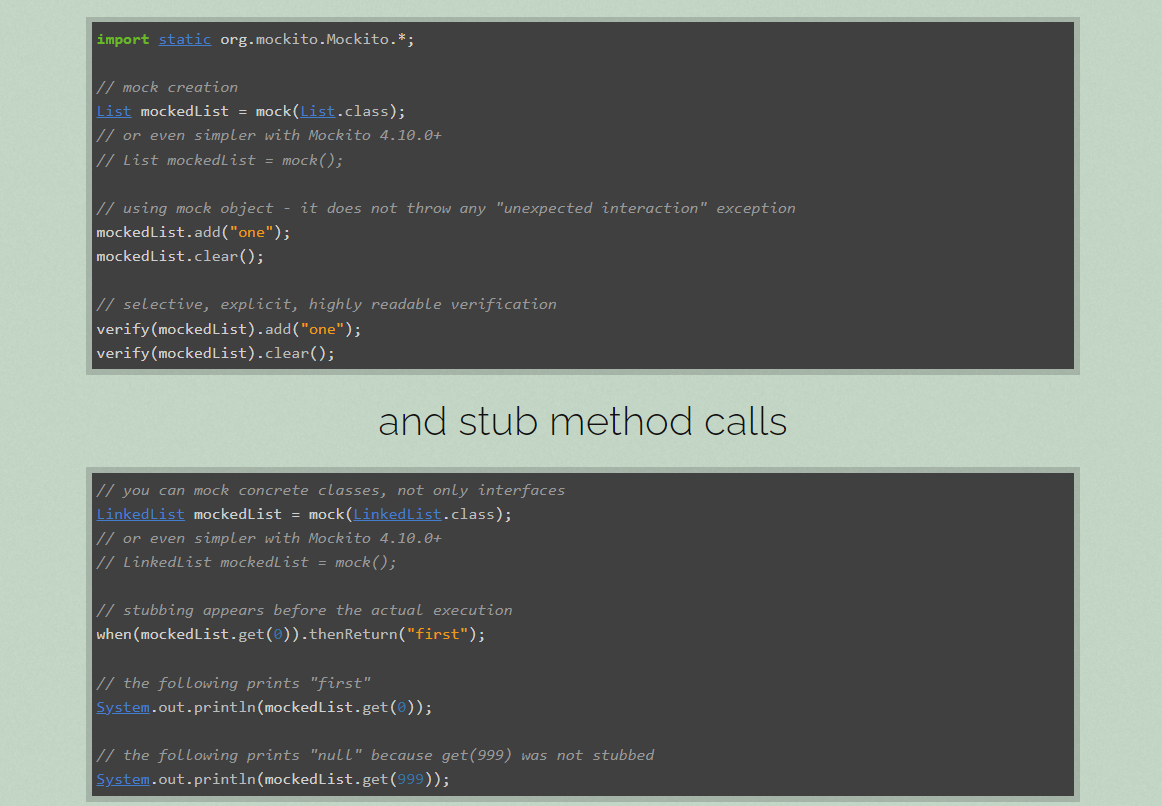
Mockito, Java için geliştirilmiş, yaygın olarak kullanılan bir mocking framework'üdür. Birim testlerinde, gerçek bağımlılıklar yerine sahte (mock) nesneler kullanarak test yapmayı kolaylaştırır. Bu sayede, testlerin daha izole ve hızlı olması sağlanır.

**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

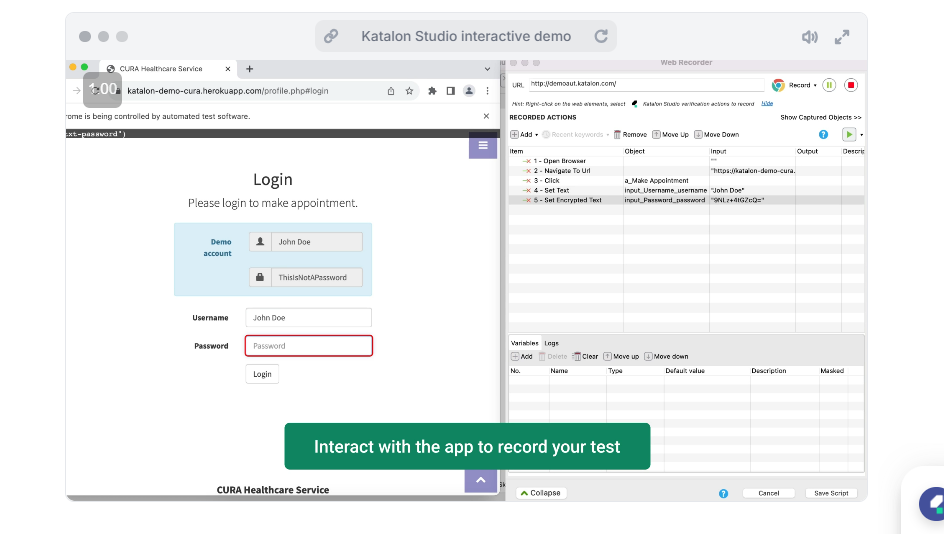
1. Katalon
2. Postman
3. SoapUI
4. Citrus
5. Selenium
6. IBM Rational Integration Tester
7. Leapwork
8. Tessy
9. LDRA
10. TestSigma
11. Katalon

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.

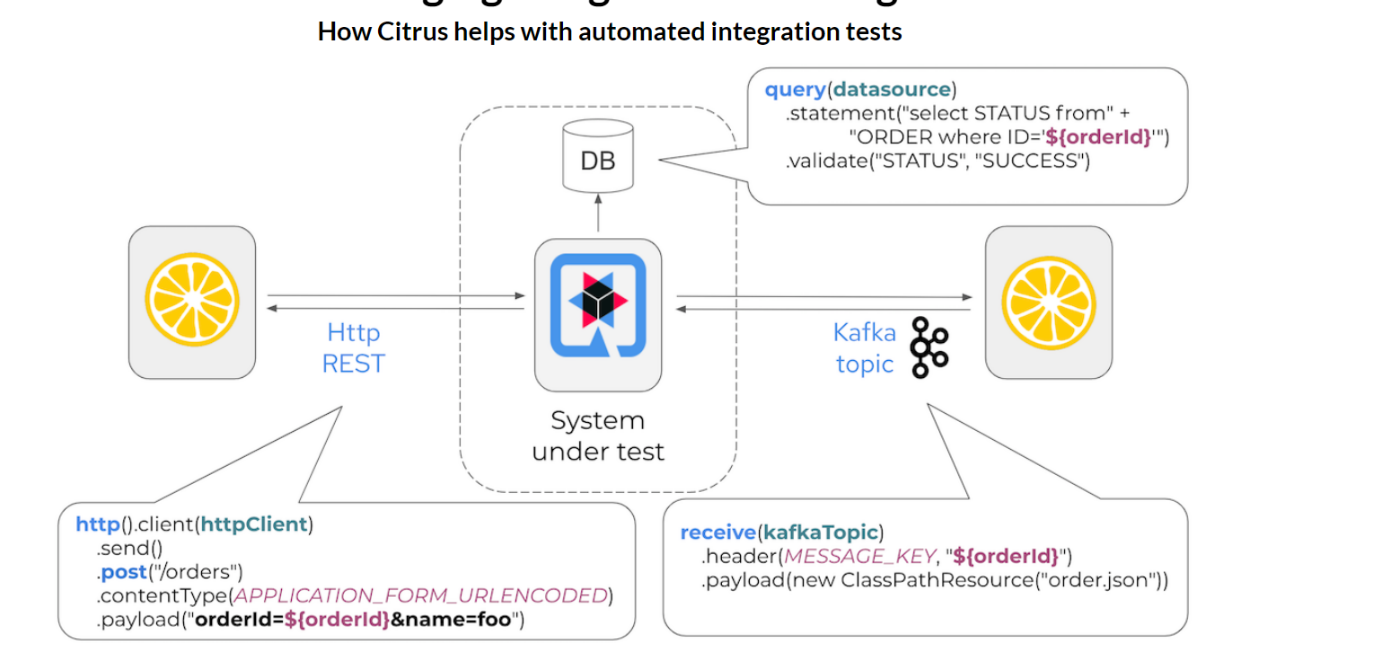
CI/CD boru hattında otomatik test yürütme

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.

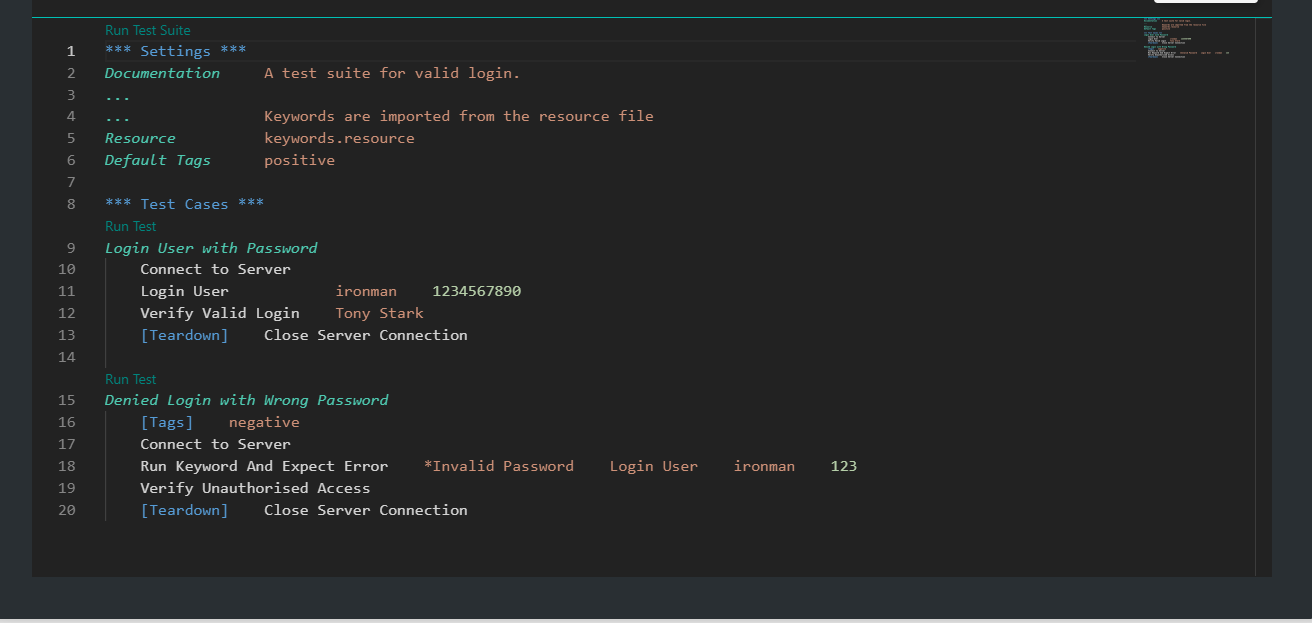
https://www.leapwork.com/test-automation

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/>

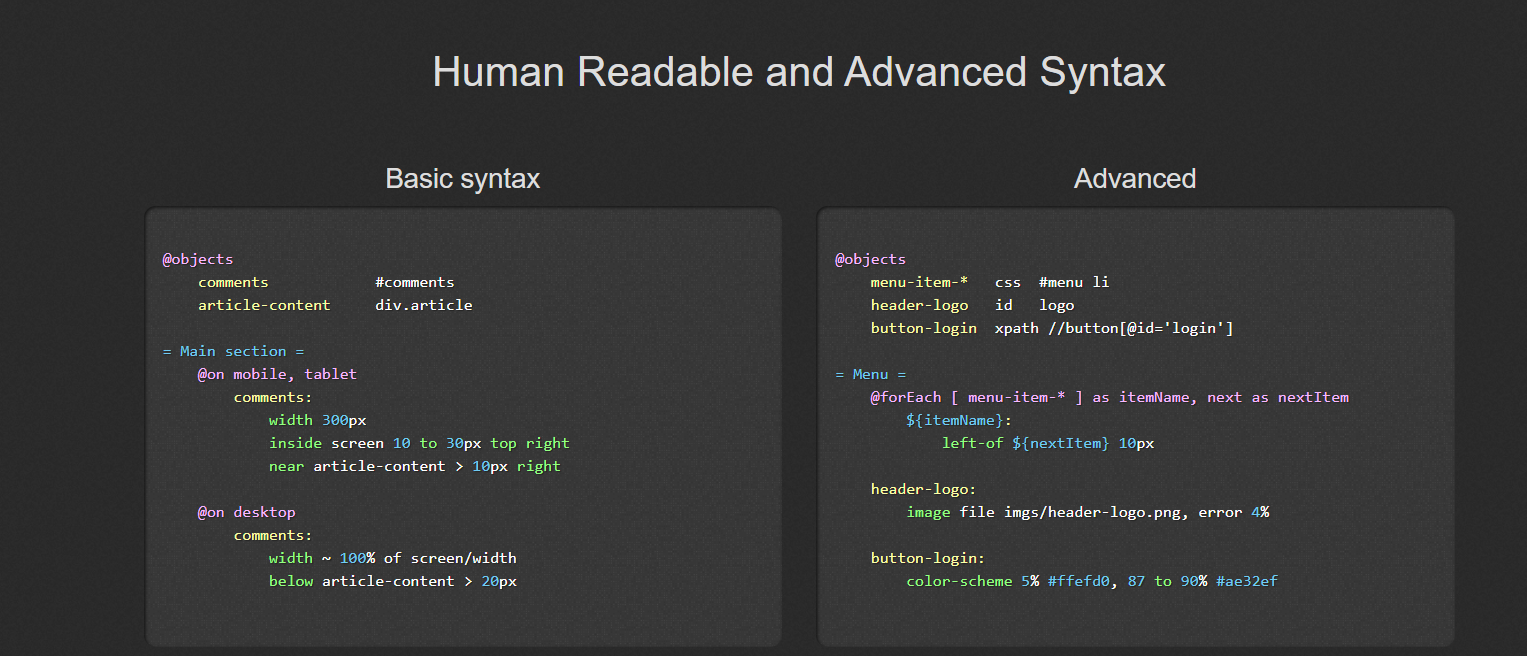
2.JMeter

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.

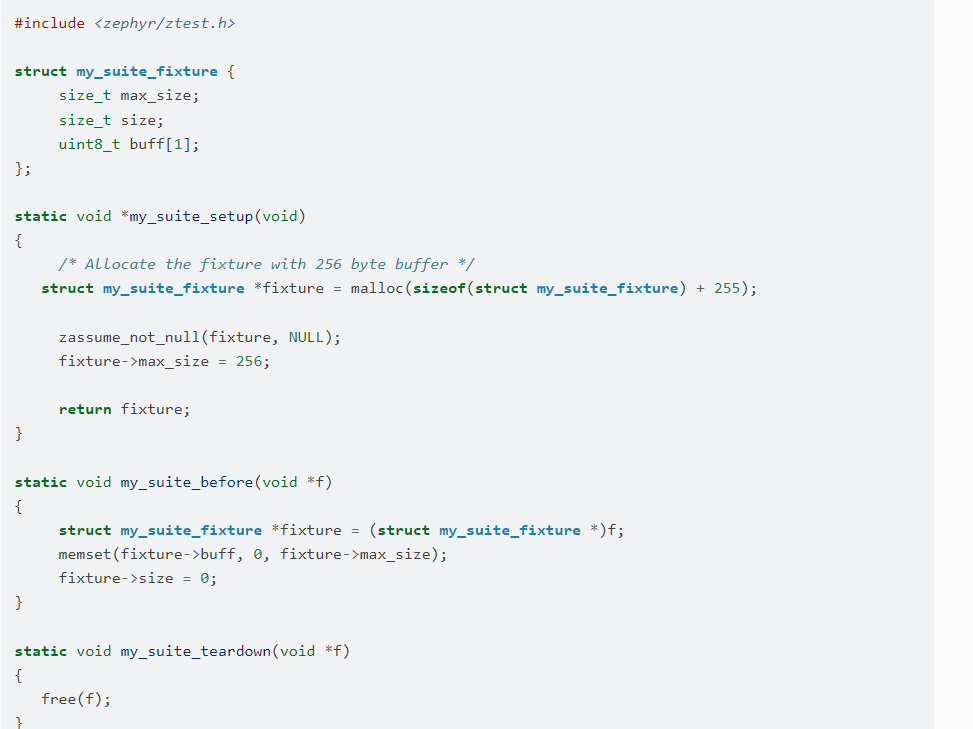
Entegrasyonlar: Zephyr, JIRA, Confluence ve Bamboo dahil olmak üzere çeşitli diğer test araçları ve platformlarıyla entegre olarak test faaliyetlerini genel yazılım geliştirme iş akışınıza entegre etmeyi kolaylaştırır.Gerçek zamanlı işbirliği: Birden fazla kullanıcının aynı test vakası üzerinde aynı 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/>

5 . Fonksiyonel Test

* 1. Selenium

 **Web Tarayıcılarıyla Entegrasyon**: Selenium WebDriver, farklı web tarayıcılarını (Chrome, Firefox, Edge, Safari vb.) otomatik olarak kontrol etmek için kullanılır. Bu, uygulamanızın farklı tarayıcılarda nasıl davrandığını test etmenizi sağlar.

 **Farklı Programlama Dili Desteği**: Selenium, Java, C#, Python, Ruby, JavaScript gibi popüler programlama dilleri ile entegre olabilir. Bu, test senaryolarınızı tercih ettiğiniz dilde yazmanıza olanak tanır.

 **Paralel Test Çalıştırma**: Selenium, test senaryolarınızı paralel olarak çalıştırmanıza izin verir. Bu, test sürelerini kısaltabilir ve test verimliliğini artırabilir.

 **Kapsamlı Element İşlemleri**: Selenium, web sayfalarındaki elementlere (butonlar, formlar, metin kutuları vb.) kolay erişim sağlar ve bu elementler üzerinde çeşitli işlemler yapılmasını mümkün kılar.

 **Esneklik ve Genişletilebilirlik**: Selenium, test senaryolarınızı dinamik ve esnek bir şekilde oluşturmanıza olanak tanır. Ayrıca, genişletilebilir bir yapıya sahiptir, yani özelleştirilmiş test senaryoları oluşturabilirsiniz.

 **Açık Kaynaklı**: Selenium, açık kaynaklı bir projedir ve ücretsiz olarak kullanılabilir. Bu, yazılım geliştirme maliyetlerini düşürebilir.

**Avantajlar:**

1. **Kapsamlı Web Uygulama Desteği**: Selenium, modern web teknolojilerini (AJAX, JavaScript, HTML5 gibi) destekler. Bu, karmaşık web uygulamalarını etkili bir şekilde test etmenizi sağlar.
2. **Cross-browser Testing**: Farklı tarayıcılarda ve platformlarda uygulamanın tutarlılığını test etmek için idealdir. Böylece uygulamanın geniş bir kullanıcı tabanında sorunsuz çalıştığından emin olabilirsiniz.
3. **Test Otomasyonu Kolaylığı**: Selenium, test senaryolarınızı otomatize etmek için güçlü bir araçtır. Bu, tekrar eden testleri manuel olarak yapmak yerine otomatikleştirebilir ve zaman kazanabilirsiniz.
4. **Topluluk Desteği ve Ekosistem**: Selenium'un geniş bir kullanıcı topluluğu vardır ve bu, sorunlarınıza hızlı çözümler bulmanıza yardımcı olabilir. Ayrıca, Selenium ekosistemi, farklı entegrasyonlar ve eklentilerle zenginleştirilmiştir.
5. **CI/CD Entegrasyonu**: Selenium testlerini sürekli entegrasyon ve dağıtım (CI/CD) süreçlerine entegre etmek kolaydır. Bu da yazılım geliştirme sürecinin hızını ve güvenilirliğini artırabilir.

**Dezvantajlar:**

 **Yüksek Başlangıç Eşiği**: Selenium'u etkili bir şekilde kullanabilmek için programlama bilgisine ve web teknolojilerine iyi bir anlayışa sahip olmak gereklidir. Başlangıçta öğrenme eğrisi olabilir.

 **Yavaşlığı**: Selenium testleri, özellikle büyük ve karmaşık web uygulamalarında yavaş olabilir. Tarayıcıların açılması, elementlerin bulunması ve işlemlerin yapılması zaman alabilir.

 **Tarayıcı Bağımlılığı**: Selenium WebDriver, farklı tarayıcılarla entegre olmak için tarayıcı sürücülerine (driver) ihtiyaç duyar. Bu sürücülerin yönetimi ve güncel tutulması ek iş yükü oluşturabilir.

 **Zorunlu Güncellemeler**: Web tarayıcıları sık sık güncellendiğinde, Selenium WebDriver sürücüleri de bu güncellemelere ayak uydurmalıdır. Bu da zaman zaman uyumsuzluk problemlerine neden olabilir.

 **Karmaşık Senaryoların Yönetimi**: Selenium ile karmaşık ve büyük test senaryolarını yönetmek zor olabilir. Test senaryolarının düzenli tutulması ve bakımı zaman alabilir.

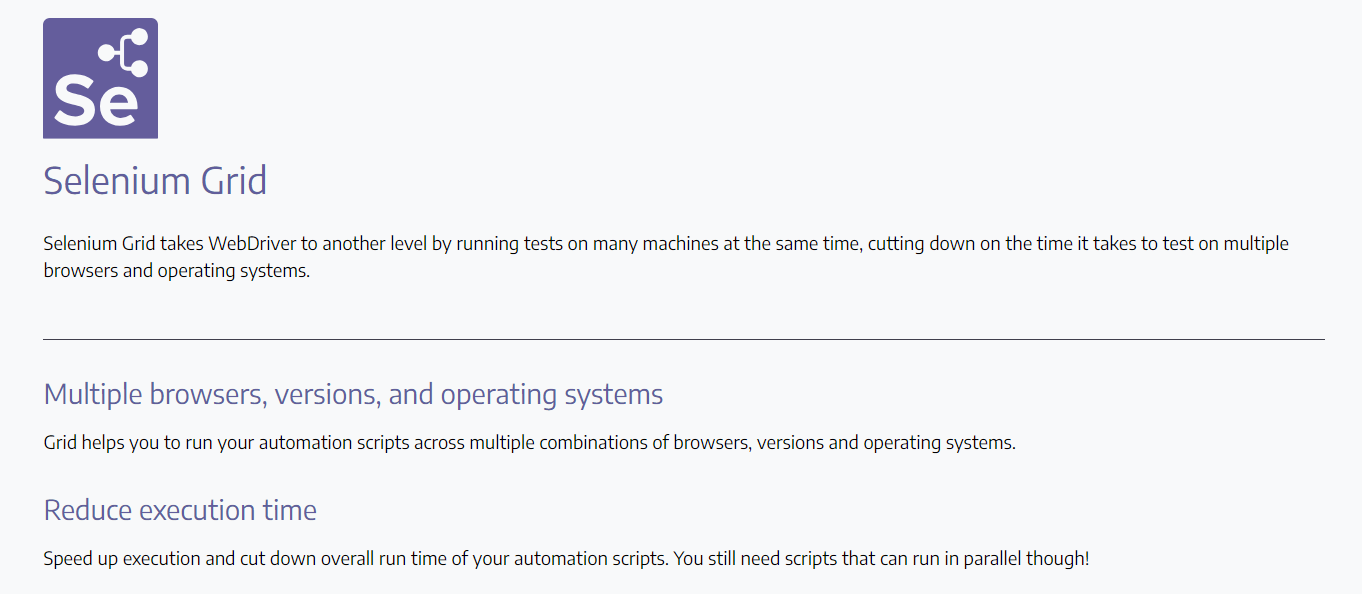
 **Mobil Uygulamalar İçin Kısıtlamalar**: Selenium, mobil uygulamaların testi için doğrudan destek sağlamaz. Bu durumda Appium gibi başka bir araca ihtiyaç duyulabilir.

 **Görsel Testler İçin Sınırlamalar**: Selenium'un doğrudan görsel test otomasyonu yetenekleri sınırlıdır. Bu tür testler için ek araçlar gerekebilir.

 **Açık Kaynaklı Destek**: Açık kaynaklı bir proje olması avantaj sağlasa da, bazı kullanıcılar için ticari destek ve hızlı çözüm talepleri karşılamada kısıtlamalar olabilir.

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

<https://github.com/arquivo/functional-tests> Githubtan örnekler



2.Appium

Appium, mobil uygulamaların test edilmesi için geliştirilmiş bir test otomasyon aracıdır. Açık kaynaklı olması, farklı mobil platformlar (Android ve iOS) için tek bir API seti üzerinden testlerin yapılabilmesine olanak tanır. Bu da geliştiricilerin aynı test senaryolarını farklı mobil platformlarda kullanabilmesini sağlar. Çok yaygın bir uygulamadır

**Özellikler:**

1. **Platform Bağımsızlık**: Appium, Android ve iOS platformlarını destekler. Bu sayede tek bir test senaryosu ile her iki platformda da uygulamanın test edilmesi mümkündür.
2. **Dil Bağımsızlık**: Appium, test senaryolarını farklı programlama dilleriyle yazmanıza olanak tanır. Java, Python, Ruby, JavaScript (Node.js), C#, PHP gibi dillerle entegre olabilir.
3. **Açık Kaynaklı Olması**: Appium, açık kaynaklı bir proje olup ücretsiz olarak kullanılabilir. Bu, geliştirme maliyetlerini düşürür ve geniş bir kullanıcı topluluğuyla desteklenir.
4. **Web ve Hybrid Uygulamaları Destekleme**: Appium, web ve hybrid mobil uygulamaların test edilmesini de destekler. Bu sayede farklı uygulama türlerinin test edilmesinde kullanılabilir.
5. **Native Uygulama Testleri**: Android ve iOS için native uygulamaların test edilmesinde etkilidir. Appium, bu uygulamaların UI ve işlevselliğini kontrol edebilir.
6. **CI/CD Entegrasyonu**: Appium, sürekli entegrasyon ve dağıtım (CI/CD) süreçlerine entegre edilebilir. Bu sayede otomatik testlerin ve sürüm dağıtımlarının otomatikleştirilmesine olanak tanır.

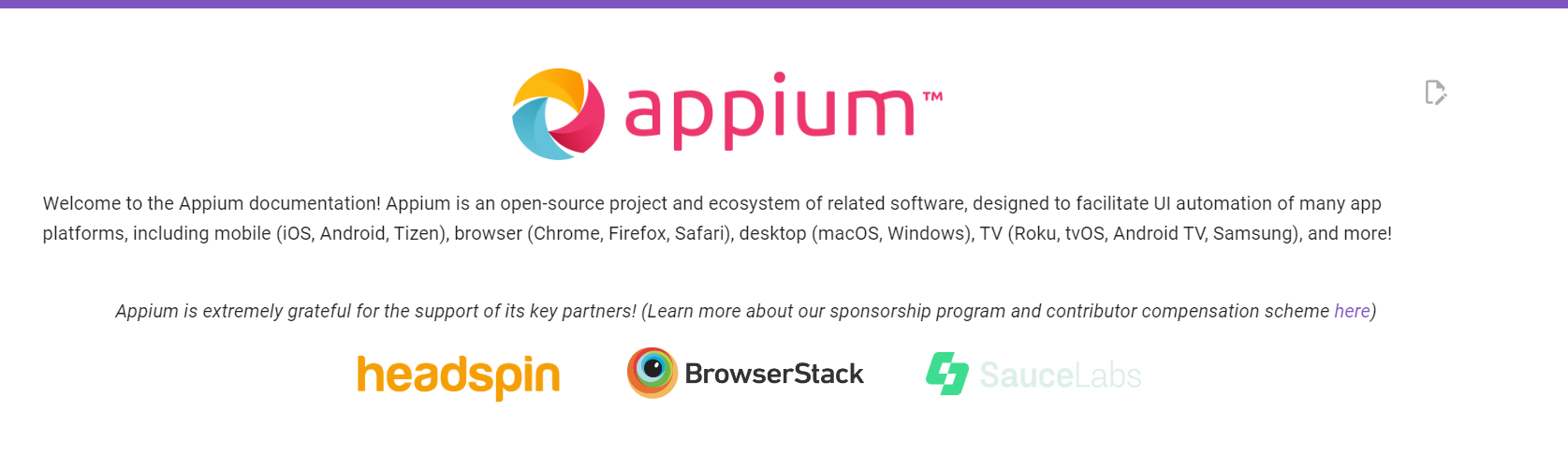
**Avantajları:**

* **Mobil Uygulamalar İçin Uygunluk**: Özellikle mobil uygulamaların test edilmesi için geliştirilmiş olup, bu alanda yaygın olarak kullanılan bir araçtır.
* **Açık Kaynaklı Olması**: Ücretsiz olarak kullanılabilir olması, geliştirme maliyetlerini düşürür.
* **Çeşitli Programlama Dilleri ile Entegrasyon**: Farklı programlama dilleriyle entegre olabilmesi, geliştiricilere esneklik sağlar.

**Dezavantajları:**

* **Yavaşlık**: Mobil uygulamaların test edilmesi genellikle daha yavaş olabilir, özellikle büyük ve karmaşık uygulamalarda.
* **Konfigürasyon Karmaşıklığı**: Başlangıçta konfigürasyon ve setup işlemleri biraz karmaşık olabilir.
* **Platform Sürümlerine Bağımlılık**: Appium, mobil platformların yeni sürümlerine uygun hale getirilmesi gereken güncellemelere ihtiyaç duyabilir.

Appium, mobil uygulama testlerinde güçlü ve esnek bir araç olup, mobil uygulama geliştiricileri ve test mühendisleri tarafından geniş çapta kullanılmaktadır.



<https://github.com/Amsterdam/amsterdam-app-appium-functional-tests> github örnek

<https://github.com/appium/appium> kullanımı

<https://appium.io/docs/en/latest/> Website

3.Cypress

Cypress, web uygulamalarının test edilmesi için tasarlanmış modern bir test otomasyon aracıdır. JavaScript tabanlı olup, geliştiricilerin web uygulamalarının ön uçlarını ve arka uçlarını test etmelerine olanak tanır. Cypress, özellikle end-to-end (E2E) testler için güçlü bir çözüm sunar ve kullanımı kolay bir arayüze sahiptir.

**Özellikleri:**

1. **JavaScript Tabanlı**: Cypress, JavaScript ile yazılan test senaryolarını destekler. Bu, JavaScript bilen geliştiriciler için öğrenmesi ve kullanması kolay bir araç yapar.
2. **Entegre Bir Çalışma Ortamı**: Cypress, test yazma, çalıştırma ve sonuçları gözlemleme gibi işlemleri entegre bir şekilde sunar. Geliştiriciler için kullanımı rahat bir arayüz sağlar.
3. **Gelişmiş Otomatikleştirme ve Debug İmkanları**: Cypress, testlerin otomatize edilmesini sağlarken aynı zamanda detaylı hata ayıklama (debugging) imkanları sunar. Testlerin adım adım nasıl çalıştığını görsel olarak izleme özelliği vardır.
4. **Tekrarlanabilir Testler**: Cypress, web uygulamalarında tekrarlanabilir E2E testlerin yazılmasını ve yönetilmesini kolaylaştırır.
5. **Time Travel (Zaman Yolculuğu) Debugging**: Cypress'in zaman yolculuğu debugging özelliği, test senaryolarının her adımında uygulamanın durumunu geriye dönebilme imkanı sağlar. Bu, hata ayıklama süreçlerini hızlandırabilir.
6. **Modern Web Teknolojilerini Destekleme**: Cypress, modern web teknolojilerini (React, Angular, Vue gibi) destekler ve bu teknolojilerle geliştirilen uygulamaları etkin bir şekilde test edebilir.
7. **CI/CD Entegrasyonu**: Cypress, sürekli entegrasyon (CI) ve sürekli dağıtım (CD) süreçlerine kolayca entegre edilebilir. Bu, otomatik testlerin ve sürüm dağıtımlarının güvenli bir şekilde yönetilmesine olanak tanır.

**Avantajları:**

* **Kullanım Kolaylığı**: Cypress, entegre bir geliştirme ortamı sunar ve test senaryolarının yazılması, çalıştırılması ve sonuçlarının görüntülenmesi için kullanıcı dostu bir arayüz sağlar.
* **Hızlı ve Güvenilir Testler**: Cypress, paralel test çalıştırma ve hızlı test sonuçları elde etme konularında güçlüdür.
* **Modern Web Uygulamaları İçin Optimizasyon**: Cypress, modern JavaScript framework'leri ve teknolojileri destekler, bu da geliştiricilerin yeni teknolojileri benimseyerek test etmelerine olanak tanır.

**Dezavantajları:**

* **Sadece Web Uygulamaları**: Cypress, sadece web uygulamalarının test edilmesi için tasarlanmıştır. Diğer türlerde (örneğin mobil uygulamalar) kullanılamaz.
* **Tam JavaScript Bilgisi Gereksinimi**: Cypress, JavaScript ile yazılan test senaryolarını destekler. Dolayısıyla, ekipte JavaScript bilgisine sahip olma gerekliliği vardır.
* **Tarayıcı Kısıtlamaları**: Cypress, sadece Chromium tabanlı tarayıcıları destekler. Diğer tarayıcılarla kullanımı sınırlıdır.

Cypress, modern web uygulamalarının test edilmesi için güçlü ve kullanıcı dostu bir araç olup, özellikle JavaScript tabanlı projelerde tercih edilmektedir. <https://www.cypress.io/> metin, ekran görüntüsü, diyagram, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

### **4.Puppeteer**

Puppeteer, Node.js ortamında çalışan bir JavaScript kütüphanesidir ve tarayıcı tabanlı otomasyon için tasarlanmıştır. Özellikle Chromium tabanlı tarayıcıları (Chrome ve Microsoft Edge gibi) kontrol etmek için kullanılır. Puppeteer, tarayıcı üzerinde sayfa gezintisi yapabilir, elementlere erişebilir, form doldurabilir, tıklama ve gezinti gibi kullanıcı etkileşimlerini simüle edebilir. Bu özellikleri sayesinde web uygulamalarının test edilmesinde ve otomatize edilmiş işlemlerin yürütülmesinde kullanılır.

### Özellikleri:

1. **Chromium Tabanlı Tarayıcıları Destekler**: Puppeteer, Chromium tabanlı tarayıcıları kontrol etmek için tasarlanmıştır. Bu tarayıcılar arasında Google Chrome ve Microsoft Edge bulunur.
2. **Hızlı ve Güvenilir Performans**: Puppeteer, tarayıcı ile iletişimde sağladığı düşük seviyeli API'lar sayesinde hızlı ve güvenilir performans sunar.
3. **Etkileşimli Kullanıcı Senaryolarını Simüle Eder**: Puppeteer, tarayıcı üzerinde gerçek kullanıcı etkileşimlerini simüle edebilir. Bu sayede form doldurma, tıklama, sayfa gezintisi gibi işlemleri otomatikleştirebilir.
4. **Ekran Görüntüsü ve PDF Oluşturma**: Puppeteer, tarayıcı sayfalarının ekran görüntülerini alabilir ve PDF dosyaları oluşturabilir.
5. **DevTools Protokolüne Erişim**: Puppeteer, tarayıcı üzerinde gelişmiş ayarlamalar yapmak için Chrome DevTools Protokolü'ne (CDP) doğrudan erişim sağlar.
6. **Headless Mod Desteği**: Puppeteer, tarayıcıyı görsel arayüz olmadan (headless) çalıştırabilir. Bu özellik, otomatik testlerin veya arka planda çalışan işlemlerin performansını artırır.
7. **Test Otomasyonu ve Web Scraping İçin Idealdir**: Puppeteer, web uygulamalarının test edilmesi ve web scraping gibi otomatize edilmiş işlemler için ideal bir seçenektir.

### Avantajları:

* **Kolay Kullanım**: Node.js ortamında çalışır ve JavaScript ile yazılan test senaryolarını destekler.
* **Güçlü API ve Fonksiyonlar**: Tarayıcı üzerinde geniş bir işlevsellik sunar ve detaylı kontrol imkanı sağlar.
* **Hızlı ve Güvenilir Performans**: Performansı yüksek ve güvenilirdir.

### Dezavantajları:

* **Sadece Chromium Tabanlı Tarayıcıları Destekler**: Diğer tarayıcılarla uyumluluk sınırlıdır.
* **Tarayıcı Sürümleri ile Uyum Sorunları**: Puppeteer, tarayıcı sürümleri ile uyumlu olmalıdır ve bu sürümlerin güncel tutulması gereklidir.

Puppeteer, özellikle Node.js tabanlı projelerde web tarayıcı tabanlı otomasyon için güçlü ve esnek bir çözüm sunar.

<https://github.com/puppeteer/puppeteer>

<https://pptr.dev/>



6.Performans Testi

* 1. Apache JMeter

Apache JMeter, açık kaynaklı bir performans test aracıdır ve özellikle web uygulamalarının performansını test etmek için kullanılır.

Apache JMeter, açık kaynaklı bir performans test aracıdır ve özellikle web uygulamalarının performansını test etmek için kullanılır. İşte Apache JMeter'ın performans testi için avantajları, dezavantajları ve özellikleri:

### Avantajları:

1. **Açık Kaynaklı ve Ücretsiz**: Apache JMeter, açık kaynaklı bir proje olup ücretsiz olarak kullanılabilir. Bu, yazılım geliştirme maliyetlerini düşürebilir.
2. **Çeşitli Protokolleri ve Teknolojileri Destekler**: JMeter, HTTP, HTTPS, FTP, JDBC, SOAP, REST, LDAP gibi farklı protokollerle ve teknolojilerle entegre olabilir. Bu, çeşitli uygulama türlerinin ve sistemlerin performans testlerinin yapılabilmesini sağlar.
3. **Kapsamlı ve Esnek Test Senaryoları**: JMeter, karmaşık test senaryolarını destekler ve farklı kullanım durumlarını simüle etmek için geniş özelleştirme imkanları sunar.
4. **Grafiksel Kullanıcı Arayüzü ve Kolay Öğrenilebilirlik**: JMeter, grafiksel bir kullanıcı arayüzüne sahiptir ve test planlarının oluşturulması ve yönetilmesi kolaydır. Bu, yeni kullanıcıların hızlıca öğrenip kullanmasını sağlar.
5. **Paralel Test Çalıştırma ve Yük Testi Desteği**: JMeter, testlerin paralel olarak çalıştırılmasını destekler ve yük testi (load testing), stres testi (stress testing), dayanıklılık testi (endurance testing) gibi senaryoları kolayca yapabilir.
6. **Raporlama ve Analiz Özellikleri**: JMeter, test sonuçlarını detaylı raporlar halinde sunar ve performans metriklerini analiz etmek için grafiksel araçlar sağlar.

### Dezavantajları:

1. **Yüksek Kaynak Tüketimi**: Büyük ölçekli testlerde, JMeter'ın sistem kaynakları (CPU, bellek) yoğun olarak kullanabilir.
2. **Karmaşık Konfigürasyon ve Ayarlar**: Karmaşık test senaryoları için ayarların doğru yapılması gerekebilir ve bu konfigürasyon süreci zaman alabilir.
3. **GUI Tabanlı Arayüzün Performansı**: Grafiksel kullanıcı arayüzü bazı durumlarda performansı etkileyebilir, özellikle büyük test planları ve yüksek talep durumlarında.
4. **Eğitim ve Öğrenme Eğrisi**: JMeter, bazı karmaşık özelliklere sahip olabilir ve başlangıçta öğrenme eğrisi olabilir.
5. **Mobil Uygulama Testleri için Sınırlı Destek**: JMeter, mobil uygulamaların test edilmesi için doğrudan destek sağlamaz. Bu durumda farklı araçlara ihtiyaç duyulabilir.

### Özellikleri:

* **HTTP, HTTPS, FTP, JDBC, SOAP, REST, LDAP gibi çoklu protokollerin desteklenmesi**
* **Modüler yapı ve eklentilerle genişletilebilirlik**
* **Farklı kullanım senaryolarını simüle etme imkanı**
* **Performans testlerinin paralel ve dağıtık olarak yapılabilmesi**
* **Grafiksel raporlama ve metrik analiz özellikleri**

Apache JMeter, geniş özellik yelpazesi ve esnek yapısıyla performans testlerinin etkili bir şekilde yapılmasını sağlayan güçlü bir araçtır.

<https://jmeter.apache.org/> <https://github.com/apache/jmeter>metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

2.LoadRunner

LoadRunner, Micro Focus tarafından geliştirilen ve özellikle büyük ölçekli uygulamaların performans testleri için tasarlanmış bir test otomasyon aracıdır.

### Avantajları:

1. **Geniş Protokol Desteği**: LoadRunner, HTTP, HTTPS, SOAP, REST, LDAP, SMTP, FTP gibi birçok farklı protokolü destekler. Bu sayede çeşitli uygulama türleri ve teknolojiler üzerinde performans testleri yapılabilir.
2. **Gelişmiş Senaryo Oluşturma**: LoadRunner, karmaşık ve gerçekçi kullanım senaryolarını simüle etmek için geniş özelleştirme imkanları sunar. Bu, gerçek dünya koşullarını yansıtan testlerin yapılabilmesini sağlar.
3. **Yüksek Ölçekli Testler**: LoadRunner, büyük ölçekli testler için optimize edilmiştir ve yüksek talep altında performansını koruyabilir.
4. **Çoklu Kullanıcı Senaryoları**: LoadRunner, aynı anda binlerce kullanıcıyı simüle edebilir ve uygulamanın yük altında nasıl performans gösterdiğini değerlendirebilir.
5. **Grafiksel Raporlama ve Analiz**: LoadRunner, detaylı grafiksel raporlar sunar ve performans metriklerini kolayca analiz edebilmenizi sağlar.
6. **CI/CD Entegrasyonu**: LoadRunner, sürekli entegrasyon (CI) ve sürekli dağıtım (CD) süreçlerine entegre edilebilir. Bu sayede otomatik testlerin ve performans testlerinin yönetimi kolaylaşır.
7. **Skriptleme ve Kodlama Desteği**: LoadRunner, test senaryolarının daha detaylı olarak skriptlenmesine ve özelleştirilmesine olanak tanır.

### Dezavantajları:

1. **Yüksek Lisans Maliyetleri**: LoadRunner, genellikle ticari bir ürün olduğundan kullanımı için lisans maliyetleri yüksek olabilir.
2. **Karmaşık Konfigürasyon ve Ayarlar**: Karmaşık test senaryoları için yapılandırma ve ayarlarının doğru yapılması gerekebilir.
3. **Eğitim ve Öğrenme Eğrisi**: LoadRunner, bazı karmaşık özelliklere sahip olabilir ve başlangıçta öğrenme eğrisi olabilir.
4. **Büyük Ölçekli Altyapı Gereksinimi**: Büyük ölçekli testler için uygun altyapı ve kaynakların sağlanması gerekebilir.

### Özellikleri:

* **Çoklu protokollerin (HTTP, HTTPS, SOAP, REST, LDAP, FTP vb.) desteklenmesi**
* **Yük testi, stres testi, dayanıklılık testi gibi farklı test senaryolarının desteklenmesi**
* **Gerçekçi kullanıcı senaryolarının simülasyonu**
* **Detaylı grafiksel raporlama ve performans analiz özellikleri**
* **CI/CD süreçlerine entegrasyon imkanı**

LoadRunner, geniş protokol desteği ve gelişmiş performans testi özellikleriyle büyük ölçekli ve karmaşık uygulamaların performansını test etmek için güçlü bir araçtır.

<https://medium.com/innova-tech/loadrunner-performans-testi-loadrunner-nedir-bbccd15d059f><https://www.perfmatrix.com/micro-focus-loadrunner-tutorial/>, https://github.com/Vycka/LoadRunner

3.OctaPerf

**Özellikleri ve Avantajları**:

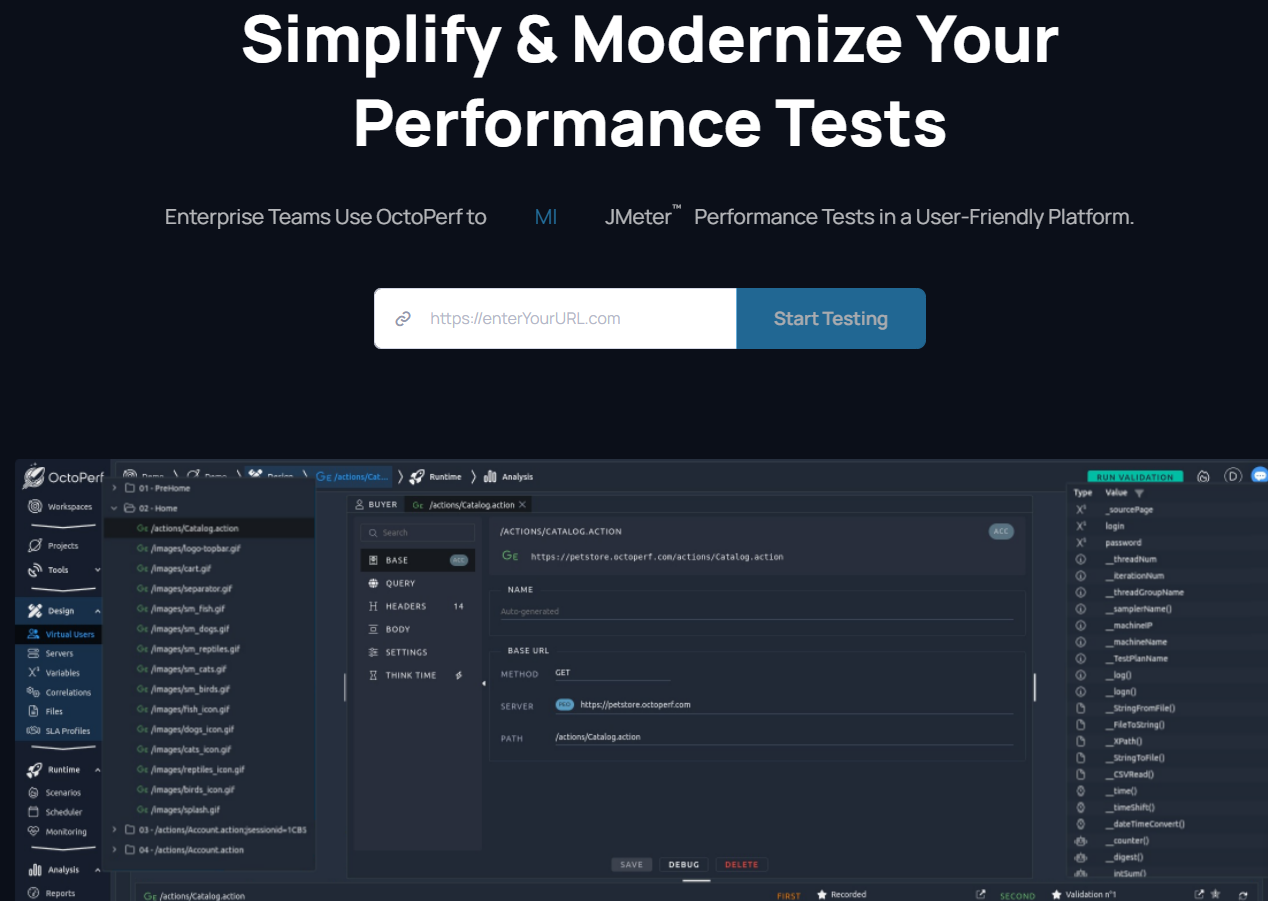
1. **JMeter Entegrasyonu**: OctaPerf, JMeter ile entegre çalışır ve kullanıcıların kodsuz bir arayüz kullanarak yeni betikler oluşturmasına veya mevcut JMX betiklerini yüklemesine olanak tanır. Bu, geleneksel JMeter'dan OctaPerf'e geçişi kolaylaştırır​
2. **Kullanıcı Dostu Arayüz**: OctaPerf, betik oluşturma sürecini önemli ölçüde basitleştiren sürükle-bırak arayüzü sunar. Bu, geleneksel yöntemlere göre çok daha kolaydır ve kullanıcıların betiklerini hızlıca düzenlemelerine olanak tanır​.
3. **Gerçek Zamanlı İzleme**: Platform, kullanıcıların test süreçlerini gerçek zamanlı olarak izlemelerine olanak tanır. Bu, anında geri bildirim almayı ve hızlıca müdahale edebilmeyi sağlar​ .
4. **Detaylı Analitik ve Raporlama**: OctaPerf, kullanıcı yükü, yanıt süresi, bağlantı süresi, gecikme, ağ süresi, veri aktarım hızı, hatalar ve isabetler gibi detaylı metrikler sunar. Bu veriler, performans sorunlarını hızlıca tespit etmeyi ve düzeltmeyi kolaylaştırır​ .

**Dezavantajları**:

1. **Kurulum Zorluğu**: OctaPerf'in bazı müşteriler için kurulum süreci zor olabilir, özellikle tamamen otomatikleştirilmiş kurulum gereksinimleri olanlar için. Bu, özellikle Rancher ve Docker gibi araçlarla entegre edilirken zorluklar yaratabilir​.
2. **Yüksek Öğrenme Eğrisi**: OctaPerf, geniş özellik yelpazesi nedeniyle başlangıçta kullanıcılar için karmaşık görünebilir. Kullanıcıların platformu etkin bir şekilde kullanabilmesi için eğitim ve belgelere ihtiyaç duyması olasıdır​ ​.
3. **Fiyatlandırma**: OctaPerf, sağladığı gelişmiş özellikler nedeniyle diğer bazı yük testi araçlarına göre daha pahalı olabilir. Özellikle küçük işletmeler veya sınırlı bütçeye sahip ekipler için maliyetli olabilir​ .

OctaPerf, detaylı analitik, kullanıcı dostu arayüz ve güçlü entegrasyon yetenekleri ile öne çıkan bir performans testi aracıdır, ancak kurulum ve öğrenme süreci bazı kullanıcılar için zorlu olabilir.

https://octoperf.com/



4.NeoLoad

NeoLoad by Tricentis, işletmeler tarafından uygulamalarının, web sitelerinin ve API'lerinin performansını, ölçeklenebilirliğini ve güvenilirliğini değerlendirmek için kullanılan popüler bir performans test aracıdır. Belli bir ücreti vardır.

Öne Çıkan Özellikler:

RealBrowser teknolojisine sahip NeoLoad, özel web ve bulut tabanlı uygulamalar için tarayıcı tabanlı performans testini birleşik bir arayüzle birleştirir.

SLA tanımı, varlık paylaşımı ve gerçek zamanlı sonuç incelemesi ile standartlaştırılmış performans testi sağlayarak ekipler arasında işbirliğini teşvik eder.

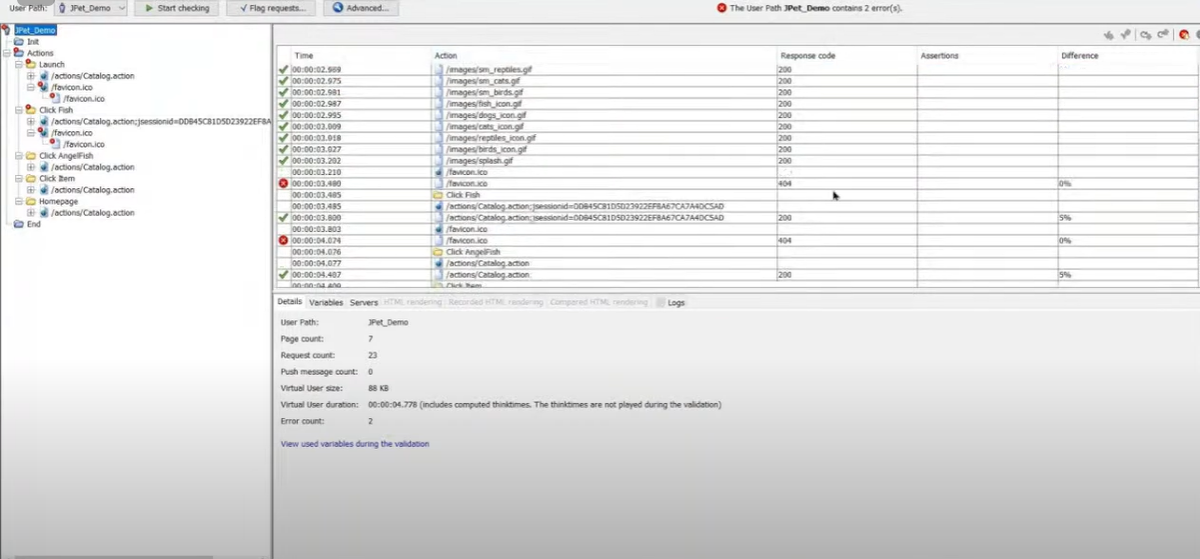
CI/CD araçlarıyla sorunsuz bir şekilde entegre olarak veya otomasyon için kullanıcı dostu CLI kullanarak performans testlerini kod olarak gerçekleştirir.

Kodlama karmaşıklığı olmadan karmaşık uygulamalar için kolayca testler oluşturur.

Otomasyon ile test komut dosyası güncellemelerini iyileştirir.

Dinamik bulut kaynağı tahsisi ile test kaynağı kullanımını optimize edin ve maliyetleri azaltın. Yük testi altyapısı rezervasyonları ile özel kaynaklar sağlayın.

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



7.Yük Testi

1.) Apache JMeter

Performans testinde de ayın tools olduğundan uzun uzun yazmıyorum.

 **Özellikler:** Ücretsiz ve açık kaynak kodludur, çok çeşitli protokolleri destekler (HTTP, HTTPS, FTP, TCP, vb.), kullanıcı dostu bir GUI'ye sahiptir, dağıtık testleri destekler.

 **Kullanım Alanları:** Web uygulamaları, web servisleri, veritabanı performans testi.

 **Web Sitesi:** <https://jmeter.apache.org/>

<https://medium.com/cloud-and-servers/jmeter-ile-rest-api-load-testi-nas%C4%B1l-yap%C4%B1l%C4%B1r-4fbea9444449> Nasıl kullanıldığına dair güzel bir kaynak.

2.Locust

Locust, açık kaynaklı bir yük test aracıdır ve web uygulamaları, mobil uygulamalar, API'ler ve diğer sistemlerin performansını test etmek için kullanılır. Locust, Python tabanlıdır ve kullanıcı davranışlarını simüle ederek sistemlerin ne kadar yük kaldırabileceğini belirlemeye yardımcı olur.

### Avantajlar

1. **Kullanım Kolaylığı:** Python tabanlı olduğu için kullanıcı dostu ve öğrenmesi kolaydır. Test senaryoları Python kodu ile yazılır.
2. **Dağıtık Test:** Birden fazla makine üzerinde test yapma yeteneğine sahiptir, bu da büyük ölçekli testleri mümkün kılar.
3. **Gerçek Zamanlı Raporlama:** Test sırasında gerçek zamanlı performans verileri sağlar ve sonuçları anında görmenizi sağlar.
4. **Esneklik:** Kullanıcı davranışlarını özelleştirmek ve karmaşık senaryolar oluşturmak için çok esnek bir yapıya sahiptir.
5. **Web Arayüzü:** Kullanımı kolay bir web arayüzü sunar, bu sayede testleri başlatmak, durdurmak ve izlemek kolaydır.

### Dezavantajlar

1. **Python Bilgisi Gerektirir:** Test senaryoları Python ile yazıldığı için Python bilmeyen kullanıcılar için başlangıçta zorlayıcı olabilir.
2. **Manuel Yapılandırma:** Dağıtık testlerde manuel yapılandırma gerekebilir, bu da bazen zaman alıcı olabilir.
3. **Yüksek Kaynak Tüketimi:** Büyük ölçekli testlerde yüksek miktarda sistem kaynağı tüketebilir.
4. **Sınırlı Grafik ve Raporlama:** Daha gelişmiş raporlama ve görselleştirme araçlarına ihtiyaç duyan kullanıcılar için yetersiz kalabilir.

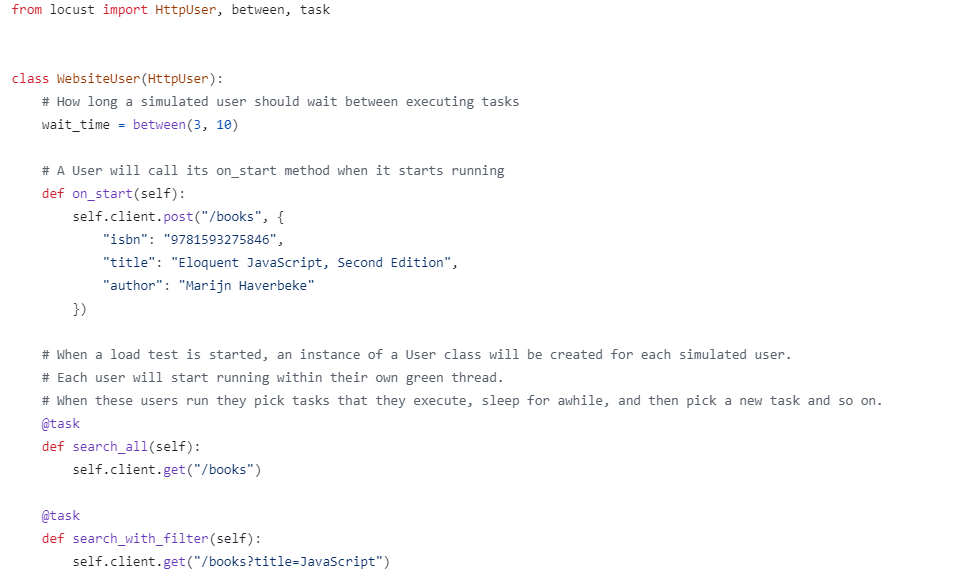
### Özellikler

1. **Kullanıcı Simülasyonu:** Kullanıcı davranışlarını kodla tanımlayarak test senaryoları oluşturma.
2. **Dağıtık Yük Testi:** Birden fazla makinada eş zamanlı yük testi gerçekleştirme.
3. **Web Arayüzü:** Testlerin başlatılması, durdurulması ve izlenmesi için web tabanlı kullanıcı arayüzü.
4. **Gerçek Zamanlı İzleme:** Test sırasında performans metriklerini gerçek zamanlı olarak izleme.
5. **Kolay Entegrasyon:** Diğer izleme ve analiz araçlarıyla entegrasyon yapabilme.

Locust, yük testi ve performans testi yapmak isteyen geliştiriciler için esnek ve güçlü bir araçtır. Hem basit hem de karmaşık senaryoları desteklemesi sayesinde geniş bir kullanıcı kitlesine hitap eder.

<https://github.com/janneri/locust-tutorial/tree/main> Nasıl kullanıldığına dair bir tutorial

<https://locust.io/> Locust Websitesi



Locust vs. JMeter

JMeter gibi araçlar, çok sayıda kaynak tüketen iş parçacığı tabanlı bir mimariye dayanmaktadır. Öte yandan Locust, daha az kaynak kullanan olay tabanlı bir yaklaşım kullanıyor

BlueConch Technologies'de teknik lider olan Rahul Solanki şöyle diyor, JMeter ile karşılaştırdığında Locust'ın kullandığı kaynak sayısının yaklaşık yüzde 70 daha azdır dedi.

"Yük jeneratörleri" terimine aşinaysanız, Locust "sürü" terimini kullanıyor - web sitenize yük bindirmek için bir çekirge sürüsünü yönlendirebileceğiniz gibi.

Her bir Locust örneğiniz için istediğiniz davranışı tanımlayabilirsiniz. Ayrıca size sürü oluşturma sürecini bir web kullanıcı arayüzünden gerçek zamanlı olarak izleme olanağı sunar.

Temel Özellikler

Test senaryolarınızı doğrudan Python kullanarak oluşturma yeteneği

Taklit etmeniz gereken kullanıcı sayısını kolayca ölçeklendirebilirsiniz

Güzel bir Web tabanlı kullanıcı arayüzüne sahiptir

Genişletilebilir

API'leri test etmek için iyi

Eksiler

Daha yeni bir araç olduğundan, JMeter'den çok daha az eklentiye sahiptir.

3.k6

k6, back-end altyapınızın performansını test etmek için geliştirici merkezli, açık kaynaklı bir yük testi aracıdır.

Performans testini sürekli entegrasyon işlem hatlarınıza dahil etmeye başlamak için de k6'yı kullanabilirsiniz.

k6 aynı zamanda Go ve JavaScript ile oluşturulmuş, çoğu geliştiricinin iş akışına iyi entegre olan Modern bir yük testi aracıdır.

Ayrıca 2023 için yeni olan K6, xk6'yı yaratıyor. xk6-browser, bir performans testi aracı olan K6 için bir uzantıdır. Tarayıcı otomasyonu ve uçtan uca web testi özellikleri sağlayarak tek bir araç kullanarak hem frond-end hem de back-end uç performansını ölçmenize olanak tanır. xk6-browser, protokol düzeyindeki testleri tarayıcı düzeyindeki testlerle birleştirerek performans testine benzersiz bir hibrit yaklaşım sunar ve web uygulamanızın performansı hakkında daha kapsamlı ve gerçekçi bir görünüm sağlar.

xk6-browser şu anda Chromium'u desteklemektedir ve gelecekte diğer tarayıcıları da desteklemeyi planlamaktadır. Sayfa öğeleriyle etkileşime girebilir, eylemleri yürütebilir ve CSS seçicileri kullanarak çeşitli kontroller gerçekleştirebilirsiniz. Araç ayrıca yüklenen DOM içeriği, ilk içerikli boyama ve ilk anlamlı boyama gibi tarayıcıya özgü performans ölçümleri de sağlar.

Temel Özellikler

Temiz komut dosyası API'si

Dağıtık ve Bulut Yürütme Sağlar

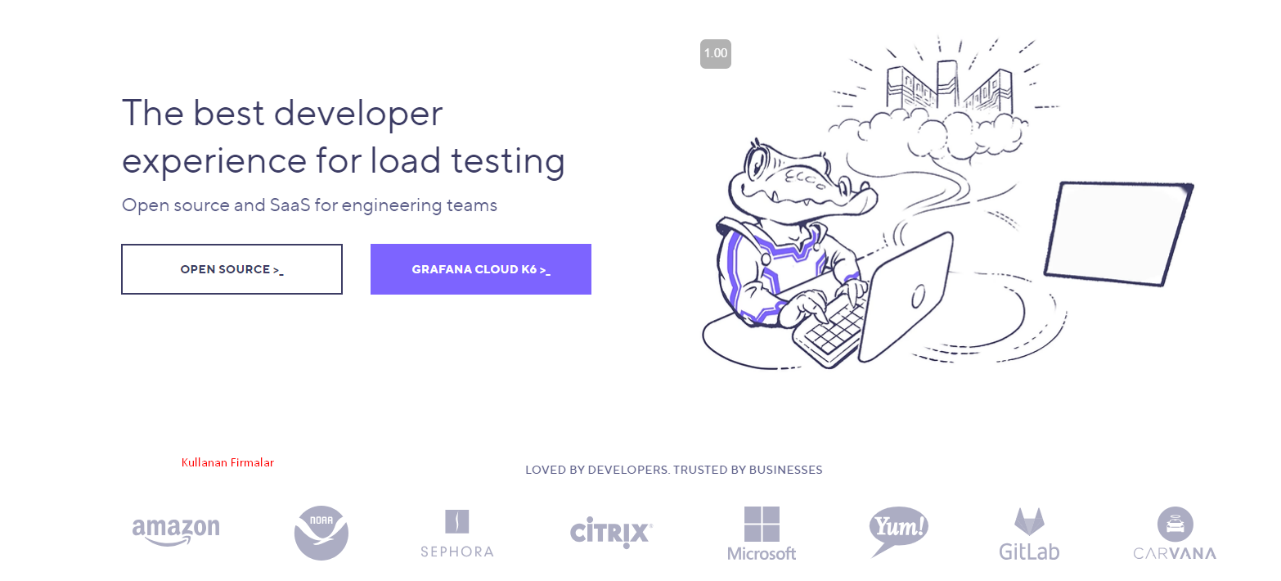
Harika REST API düzenleme yeteneği

Yeni xk6 uzantısını kullanarak tarayıcı düzeyinde frond-end performansı gerçekleştirin.

Open Soruce yapısında olduğundan dolayı ücretsizdir.

<https://medium.com/@alperen-tekin/k6-ile-y%C3%BCk-testi-33e501836364> Burada Türkçe şekilde çok güzel açıklanmış ve kullanımı kolay.

<https://k6.io/> Dokümantasyonlara buradan ulaşılabilinir.



4.Tsung

Tsung, Erlang tabanlı, açık kaynaklı bir dağıtık yük test aracıdır ve yüksek ölçekli performans testleri yapmak için kullanılır. Tsung, HTTP, WebDAV, SOAP, PostgreSQL, MySQL, LDAP, MQTT ve diğer birçok protokolü destekler.Ücretsizdir.

### Avantajlar

1. **Yüksek Ölçeklenebilirlik:** Tsung, büyük ölçekli yük testleri gerçekleştirmek için idealdir ve birden fazla makinede dağıtık test yapabilir.
2. **Çeşitli Protokol Desteği:** Birçok farklı protokolü destekler, bu da çeşitli türdeki uygulamaları test etmeyi mümkün kılar.
3. **Gerçek Zamanlı İzleme:** Test sırasında gerçek zamanlı performans verileri sunar ve sonuçları anında analiz etmenizi sağlar.
4. **Konfigürasyon Dosyaları:** XML tabanlı konfigürasyon dosyaları kullanarak test senaryoları oluşturmak mümkündür, bu da test senaryolarını anlamayı ve yönetmeyi kolaylaştırır.
5. **Yüksek Performans:** Tsung, hafif ve hızlı bir araçtır ve düşük kaynak tüketimiyle yüksek performans sunar.

### Dezavantajlar

1. **Öğrenme Eğrisi:** Tsung'u etkili bir şekilde kullanmak için bazı temel Erlang bilgisi ve XML konfigürasyon dosyaları hakkında bilgi sahibi olmak gerekebilir.
2. **Karmaşık Kurulum:** Tsung'un kurulumu ve yapılandırması bazı kullanıcılar için karmaşık olabilir.
3. **Sınırlı Topluluk Desteği:** Diğer popüler yük test araçlarına göre daha küçük bir kullanıcı topluluğuna sahip olabilir, bu da destek ve kaynak bulmayı zorlaştırabilir.
4. **Görselleştirme:** Daha gelişmiş raporlama ve görselleştirme araçlarına ihtiyaç duyan kullanıcılar için yetersiz kalabilir.

### Özellikler

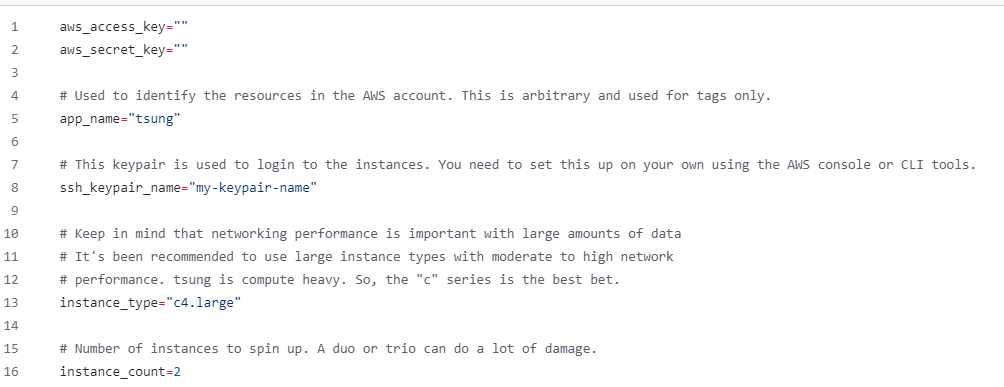
1. **Çeşitli Protokol Desteği:** HTTP, WebDAV, SOAP, PostgreSQL, MySQL, LDAP, MQTT ve diğer protokollerle uyumludur.
2. **Dağıtık Yük Testi:** Birden fazla makinada dağıtık testler yaparak büyük ölçekli yük testleri gerçekleştirebilir.
3. **Konfigürasyon Dosyaları:** Test senaryolarını XML tabanlı konfigürasyon dosyaları ile tanımlama.
4. **Gerçek Zamanlı İzleme:** Test sırasında performans metriklerini gerçek zamanlı olarak izleme ve raporlama.
5. **Esneklik:** Karmaşık test senaryolarını ve kullanıcı davranışlarını özelleştirme yeteneği.

Tsung, büyük ölçekli ve dağıtık performans testleri yapmak isteyenler için güçlü ve esnek bir araçtır. Geniş protokol desteği ve yüksek performansı sayesinde çeşitli türdeki uygulamaları test etmeye uygundur. Ancak, kullanım ve kurulum sürecindeki karmaşıklık bazı kullanıcılar için dezavantaj olabilir.

<http://tsung.erlang-projects.org/> Dokümantasyonlara ulaşabilmek için web sitesi

<https://github.com/processone/tsung> Github Linki

<https://github.com/mkornatz/tsung-pack> Bir adet örnek kullanım



1. **Kullanılabilirlik Testi**

**1.Hotjar**

Hotjar, kullanıcı deneyimi (UX) ve davranış analizi için kullanılan bir web analitik aracıdır.

### Özellikler

1. **Isı Haritaları (Heatmaps):**
   * Kullanıcıların hangi bölgelerde daha fazla etkileşimde bulunduğunu gösterir.
   * Tıklama, dokunma ve kaydırma (scroll) haritaları sunar.
2. **Kullanıcı Kayıtları (Session Recordings):**
   * Kullanıcıların site üzerindeki oturumlarını kaydeder.
   * Kullanıcıların fare hareketlerini, tıklamalarını ve kaydırmalarını izleme imkanı sağlar.
3. **Dönüşüm Huni (Conversion Funnels):**
   * Kullanıcıların belirli bir hedefe (örneğin bir satın alma) ulaşırken hangi adımlarda terk ettiğini gösterir.
   * Dönüşüm oranlarını iyileştirmek için hangi sayfaların optimize edilmesi gerektiğini belirlemeye yardımcı olur.
4. **Form Analizi:**
   * Formların hangi alanlarının kullanıcılar tarafından doldurulmadığını veya hangi alanlarda kullanıcıların zorlandığını analiz eder.
   * Form dönüşüm oranlarını artırmak için optimizasyon önerileri sunar.
5. **Anketler ve Geri Bildirim Araçları:**
   * Kullanıcılardan doğrudan geri bildirim almak için anketler oluşturulabilir.
   * Çıkış anketleri, anında geri bildirim widget’ları ve NPS (Net Promoter Score) anketleri gibi çeşitli anket türleri sunar.

### Avantajlar

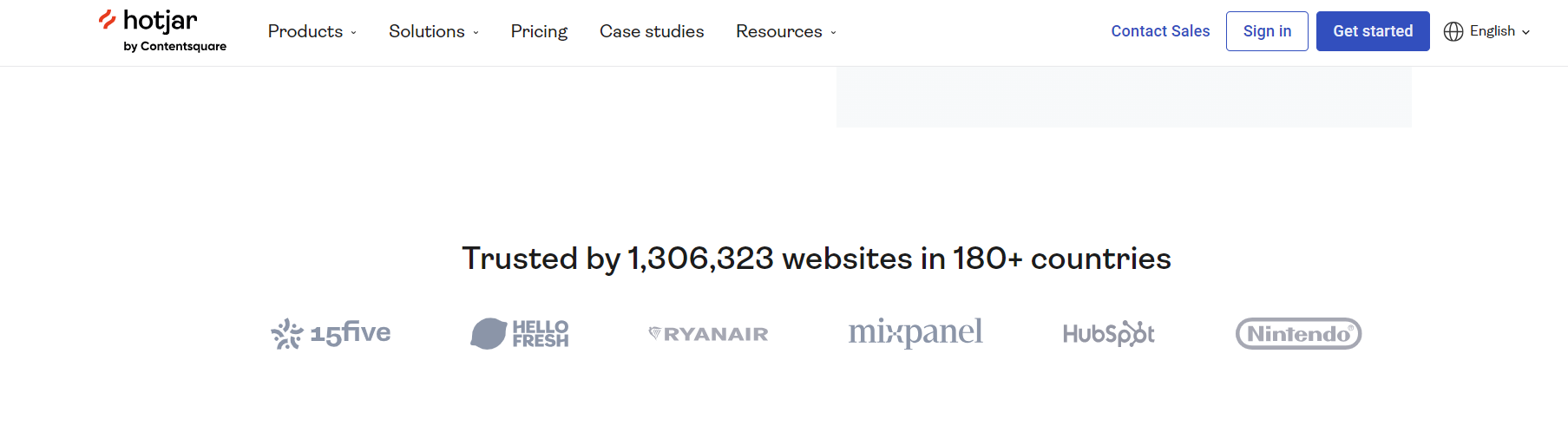
1. **Kullanıcı Davranışını Anlama:**
   * Kullanıcıların sitede nasıl davrandığını ve etkileşimde bulunduğunu anlamaya yardımcı olur.
   * Kullanıcı deneyimini iyileştirmek için veri odaklı kararlar alınmasını sağlar.
2. **Kullanıcı Geri Bildirimi:**
   * Doğrudan kullanıcıdan geri bildirim alma imkanı sunar.
   * Anket ve geri bildirim araçları, kullanıcıların yaşadığı sorunları ve beklentilerini anlamak için kullanışlıdır.
3. **Kolay Entegrasyon:**
   * Hotjar, çoğu web sitesine kolayca entegre edilebilir.
   * Kullanımı kolay arayüzü sayesinde karmaşık teknik bilgi gerektirmez.
4. **Veri Görselleştirme:**
   * Isı haritaları ve kullanıcı kayıtları gibi görsel araçlar sayesinde verilerin anlaşılması kolaydır.
   * Grafikler ve raporlar, kullanıcı davranışlarını ve eğilimlerini net bir şekilde gösterir.

### Dezavantajlar

1. **Veri Gizliliği ve Güvenlik:**
   * Kullanıcı oturum kayıtları ve geri bildirimler, veri gizliliği endişeleri yaratabilir.
   * Özellikle GDPR ve diğer veri koruma yasalarına uyum sağlanması gerekebilir.
2. **Performans Etkisi:**
   * Hotjar’ın web sitesine entegre edilmesi, site performansını biraz yavaşlatabilir.
   * Kullanıcı oturum kayıtları ve diğer özellikler, web sitesinde ek yük yaratabilir.
3. **Ücretlendirme:**
   * Hotjar’ın ücretsiz sürümü sınırlı özellikler sunar.
   * Tam özelliklerden faydalanmak için ücretli planlara geçmek gerekebilir.
4. **Analiz Sınırlamaları:**
   * Karmaşık kullanıcı yolculukları ve detaylı analizler için yetersiz kalabilir.
   * Daha derinlemesine analizler için ek araçlar gerekebilir.

Hotjar, kullanıcı deneyimi iyileştirmeleri için değerli bir araç olabilir, ancak veri gizliliği ve performans etkilerini göz önünde bulundurarak kullanılması önemlidir.

<https://www.hotjar.com/> Web sitesi



2.Maze

1. **Kullanıcı Testleri:**
   * Kullanıcıların belirli görevleri tamamlamalarını ve bu süreçte nasıl davrandıklarını analiz eder.
   * Görev tamamlama süreleri, tıklamalar ve kullanıcı akışları hakkında veri sağlar.
2. **Prototip Testleri:**
   * Figma, Sketch, Adobe XD gibi tasarım araçlarıyla oluşturulan prototipleri doğrudan test etmeye olanak tanır.
   * Tasarımların kullanıcılar tarafından nasıl etkileşimde bulunulduğunu görmeyi sağlar.
3. **Anketler ve Anlık Geri Bildirim:**
   * Kullanıcıların test sırasında veya sonrasında anketler doldurmasını sağlar.
   * Doğrudan kullanıcıdan geri bildirim toplayarak tasarımların nasıl iyileştirilebileceği konusunda içgörüler sunar.
4. **Analiz ve Raporlama:**
   * Test sonuçlarını görsel ve anlaşılır raporlar şeklinde sunar.
   * Kullanıcı davranışlarını analiz etmek için ısı haritaları ve tıklama haritaları sağlar.
5. **Görev Analizi:**
   * Kullanıcıların belirli görevleri nasıl tamamladığını ve hangi adımlarda zorlandığını analiz eder.
   * Görev tamamlama oranları ve başarı oranlarını ölçer.

### Avantajlar

1. **Kolay Entegrasyon:**
   * Popüler tasarım araçlarıyla sorunsuz entegrasyon sağlar.
   * Prototipleri doğrudan test etme ve sonuçları analiz etme imkanı sunar.
2. **Kullanıcı Davranışını Anlama:**
   * Kullanıcıların tasarımlarla nasıl etkileşime geçtiğini ve hangi alanlarda zorlandığını anlamaya yardımcı olur.
   * Kullanıcı deneyimini iyileştirmek için değerli içgörüler sağlar.
3. **Veri Odaklı Karar Verme:**
   * Test sonuçlarına dayalı veri odaklı kararlar alınmasını sağlar.
   * Tasarım sürecinde objektif geri bildirimler alarak tasarımların optimize edilmesine yardımcı olur.
4. **Zaman ve Maliyet Tasarrufu:**
   * Kullanıcı testlerini çevrimiçi olarak gerçekleştirme imkanı sunar, bu da zaman ve maliyet tasarrufu sağlar.
   * Uzaktan kullanıcı testleri sayesinde geniş bir kullanıcı kitlesine ulaşılabilir.

### Dezavantajlar

1. **Kullanıcı Seçimi:**
   * Doğru kullanıcıları testlere dahil etmek bazen zor olabilir.
   * Test sonuçlarının güvenilirliği, kullanıcı kitlesinin çeşitliliğine ve uygunluğuna bağlıdır.
2. **Test Kapsamı:**
   * Daha karmaşık ve uzun süreli kullanıcı yolculuklarını test etmek için yetersiz kalabilir.
   * Prototip testlerinin ötesinde daha derinlemesine analizler gerektiren projelerde sınırlı olabilir.
3. **Veri Gizliliği:**
   * Kullanıcıların testlerde sağladığı verilerin gizliliği ve güvenliği endişe yaratabilir.
   * Veri koruma yasalarına uyum sağlanması gerekebilir.
4. **Ücretlendirme:**
   * Maze’in tam özelliklerinden faydalanmak için ücretli planlara geçmek gerekebilir.
   * Ücretsiz sürümde sınırlı özellikler ve test imkanı sunulabilir.

Maze, kullanıcı deneyimi ve prototip testleri için güçlü bir araçtır, ancak kullanıcı seçimi ve veri gizliliği konularında dikkatli olunması önemlidir. Maze, kullanıcı davranışlarını anlamak ve tasarımları optimize etmek için değerli içgörüler sağlayabilir.

<https://maze.co/> Websitesi

****

**3.UsabilityHub**

UsabilityHub, kullanılabilirlik testleri ve kullanıcı geri bildirimleri toplamak için kullanılan çevrimiçi bir platformdur. Bu platform, tasarımcılar ve ürün yöneticileri tarafından, kullanıcıların web siteleri, uygulamalar veya diğer dijital ürünlerle nasıl etkileşimde bulunduğunu anlamak için kullanılır.

### Özellikler

1. **Beş Saniye Testi (Five Second Test):**
   * Kullanıcılara bir tasarımı beş saniye boyunca gösterir ve ardından ne hatırladıklarını sorar.
   * Tasarımın ilk izlenimlerini ve temel mesajlarının anlaşılabilirliğini ölçer.
2. **Tıklama Testi (Click Test):**
   * Kullanıcılara belirli bir görevi tamamlamalarını ister ve nerelere tıkladıklarını kaydeder.
   * Kullanıcıların tıklama davranışlarını ve tasarımın kullanım kolaylığını değerlendirir.
3. **Navigasyon Testi (Navigation Test):**
   * Kullanıcılara belirli görevler vererek sitenizde veya uygulamanızda nasıl gezindiklerini analiz eder.
   * Navigasyon yapısının etkinliğini ve kullanıcıların hedeflerine ulaşma sürecini değerlendirir.
4. **Tasarım Karşılaştırma Testi (Preference Test):**
   * Kullanıcılara iki veya daha fazla tasarım sunar ve hangisini tercih ettiklerini sorar.
   * Farklı tasarım seçenekleri arasında tercih edilenin belirlenmesine yardımcı olur.
5. **Anketler ve Geri Bildirim Formları:**
   * Kullanıcılardan doğrudan geri bildirim almak için anketler ve geri bildirim formları oluşturabilirsiniz.
   * Kullanıcıların yaşadığı zorlukları ve beklentilerini daha iyi anlamanızı sağlar.

### Avantajlar

1. **Kolay Kullanım:**
   * Kullanıcı dostu arayüzü sayesinde karmaşık teknik bilgi gerektirmez.
   * Hızlı ve etkili kullanılabilirlik testleri oluşturmayı sağlar.
2. **Gerçek Kullanıcı Geri Bildirimi:**
   * Hedef kitlenizden doğrudan geri bildirim almanızı sağlar.
   * Kullanıcı davranışlarını ve tercihlerini anlamanızı kolaylaştırır.
3. **Çeşitli Test Türleri:**
   * Farklı test türleri ile tasarımlarınızı çeşitli açılardan değerlendirebilirsiniz.
   * Her test türü, farklı kullanıcı deneyimi bileşenlerini analiz etmenizi sağlar.
4. **Veri Odaklı Karar Verme:**
   * Test sonuçlarına dayalı veri odaklı kararlar almanıza yardımcı olur.
   * Tasarım sürecinde objektif geri bildirimler alarak iyileştirmeler yapabilirsiniz.

### Kullanım Alanları

1. **Web ve Mobil Tasarımlar:**
   * Web siteleri ve mobil uygulamalar için kullanılabilirlik testleri yaparak kullanıcı deneyimini optimize edebilirsiniz.
2. **Prototip ve Mockup Değerlendirmeleri:**
   * Yeni tasarımların veya güncellemelerin kullanıcılar tarafından nasıl algılandığını görmek için prototipleri test edebilirsiniz.
3. **Navigasyon Yapısı ve Bilgi Mimarisi:**
   * Sitelerinizin veya uygulamalarınızın navigasyon yapısını ve bilgi mimarisini değerlendirerek kullanıcıların kolayca gezinebileceği bir yapı oluşturabilirsiniz.
4. **Tasarım Seçenekleri Karşılaştırması:**
   * Farklı tasarım seçeneklerini karşılaştırarak en uygun ve kullanıcı dostu olanı belirleyebilirsiniz.

### Özet

UsabilityHub, kullanılabilirlik testleri ve kullanıcı geri bildirimi toplama konusunda güçlü bir araçtır. Çeşitli test türleri ve kullanıcı dostu arayüzü sayesinde, tasarımlarınızı hızlı ve etkili bir şekilde değerlendirebilir, kullanıcı deneyimini optimize edebilirsiniz. Bu araç, web ve mobil uygulama geliştirme süreçlerinde değerli içgörüler sağlar ve veri odaklı kararlar almanıza yardımcı olur.

**9.Stres Testi**

1. **Apache JMeter**

Daha önce açıklanmıştı.

1. **K6**

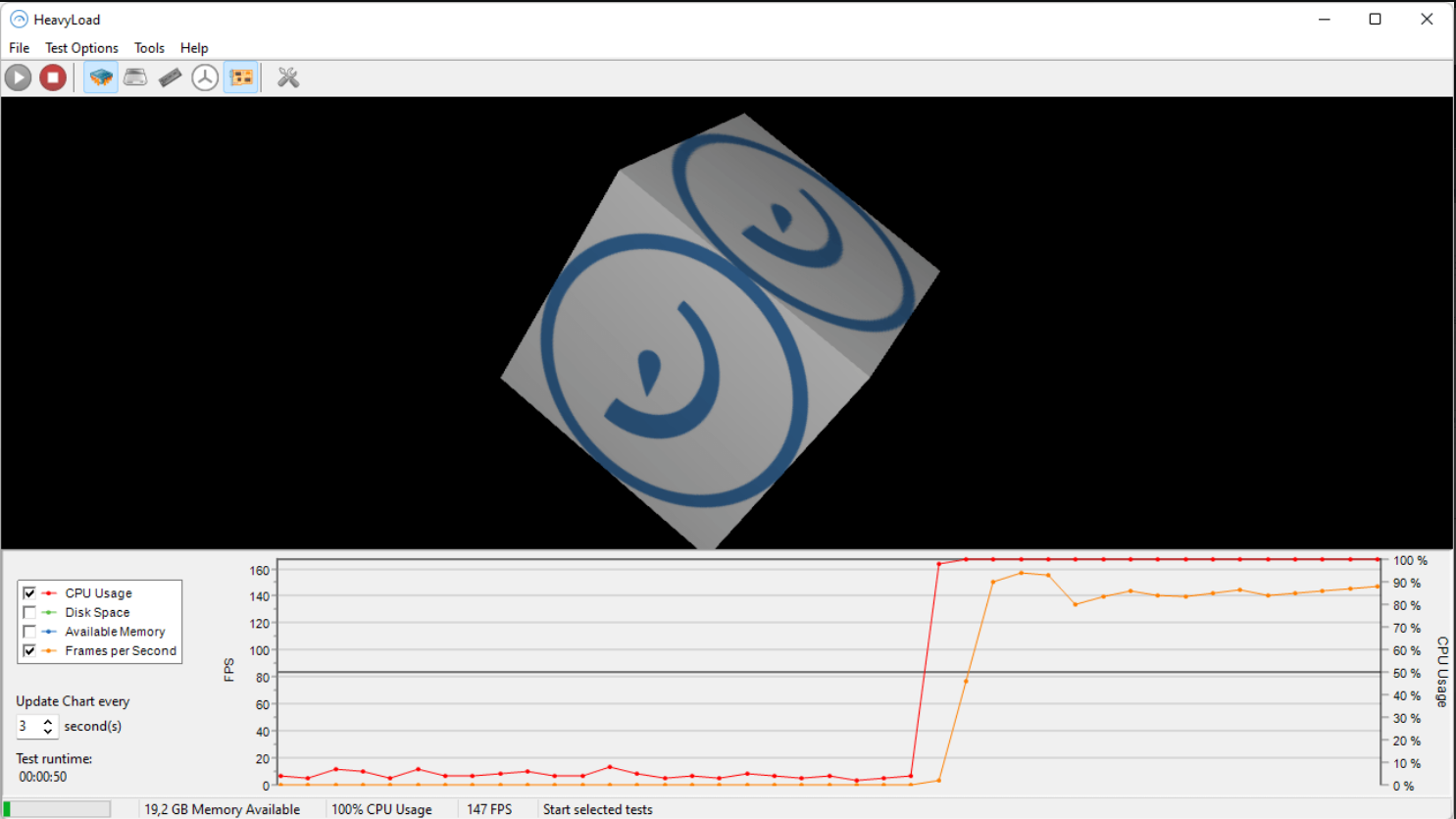
Daha Önce Açıklanmıştı

**3.HeavyLoad (Jam Software)**

Ücretsiz stres testi aracı HeavyLoad, bilgisayarınızı sınırlarına getirmek için geliştirilmiştir. Şu anda Microsoft tarafından desteklenen tüm Windows sürümlerine ve istemci/sunucu sürümlerine (32-bit ve 64-bit) kurulabilir.

HeavyLoad, iş istasyonu veya sunucu bilgisayarınızı ağır bir yük altına sokar ve hala güvenilir bir şekilde çalışıp çalışmayacaklarını test etmenizi sağlar. En Değerli Sistemleriniz için Stres TestiBilgisayar performans testimizle en önemli sistemlerinizin en yüksek yüklere dayanıp dayanamayacağını önceden kontrol edin. Tüm Sistem Bileşenleri için TestlerCPU, GPU, RAM, sabit disk veya hepsinin birlikte olması fark etmez: HeavyLoad sınırı zorlar. Kolay KullanımSezgisel grafik kullanıcı arayüzü ve görsel geri bildirim sayesinde, kıyaslama yazılımımız için uzman bilgisi gerekmez.

<https://www.jam-software.com/heavyload> Websitesi



**10.Güvenilirlik Testi**

* 1. **OWASP ZAP (Zed Attack Proxy)**
* OWASP ZAP (Zed Attack Proxy), web uygulamalarını güvenlik açıklarına karşı test etmek için kullanılan açık kaynaklı bir güvenlik aracıdır.

### Özellikler

1. **Proxy Sunucu**: ZAP, bir proxy sunucusu olarak çalışır ve bu sayede HTTP/HTTPS trafiğini kesip inceler.
2. **Otomatik Tarayıcı**: Web uygulamalarındaki güvenlik açıklarını bulmak için otomatik tarama yapar.
3. **Manuel Test Araçları**: Güvenlik uzmanlarının manuel testler yapabilmesi için çeşitli araçlar sunar.
4. **Spidering**: Web uygulamasının tüm bağlantılarını keşfederek uygulamayı tarar.
5. **Fuzzing**: Web uygulamasının girdi noktalarına rastgele veya belirli girdiler göndererek güvenlik açıklarını tespit eder.
6. **Scripting Desteği**: ZAP, özel test senaryoları oluşturmak için scripting desteği sunar.
7. **Raporlama**: Test sonuçlarını farklı formatlarda (HTML, XML, JSON) raporlar halinde sunar.
8. **Pasif Tarama**: Web trafiğini izler ve güvenlik açıklarını belirlerken uygulamayı etkilemez.
9. **Aktif Tarama**: Web uygulamasına saldırılar yaparak güvenlik açıklarını aktif olarak test eder.

### Avantajlar

1. **Açık Kaynak ve Ücretsiz**: ZAP, açık kaynaklı ve ücretsizdir, bu da onu küçük ve büyük ölçekli şirketler için erişilebilir kılar.
2. **Kullanıcı Dostu**: Kullanıcı dostu arayüzü sayesinde hem yeni başlayanlar hem de uzmanlar için uygundur.
3. **Genişletilebilirlik**: Eklentiler ve scriptler sayesinde özelleştirilebilir ve genişletilebilir.
4. **Geniş Topluluk ve Destek**: Geniş bir kullanıcı topluluğu ve aktif destek ile dokümantasyon ve forumlar üzerinden yardım alınabilir.
5. **Platform Bağımsız**: Java ile yazıldığı için farklı işletim sistemlerinde (Windows, Mac, Linux) çalışabilir.
6. **Güvenlik Açığı Veritabanı**: Sürekli güncellenen bir güvenlik açığı veritabanına sahiptir.

### Dezavantajlar

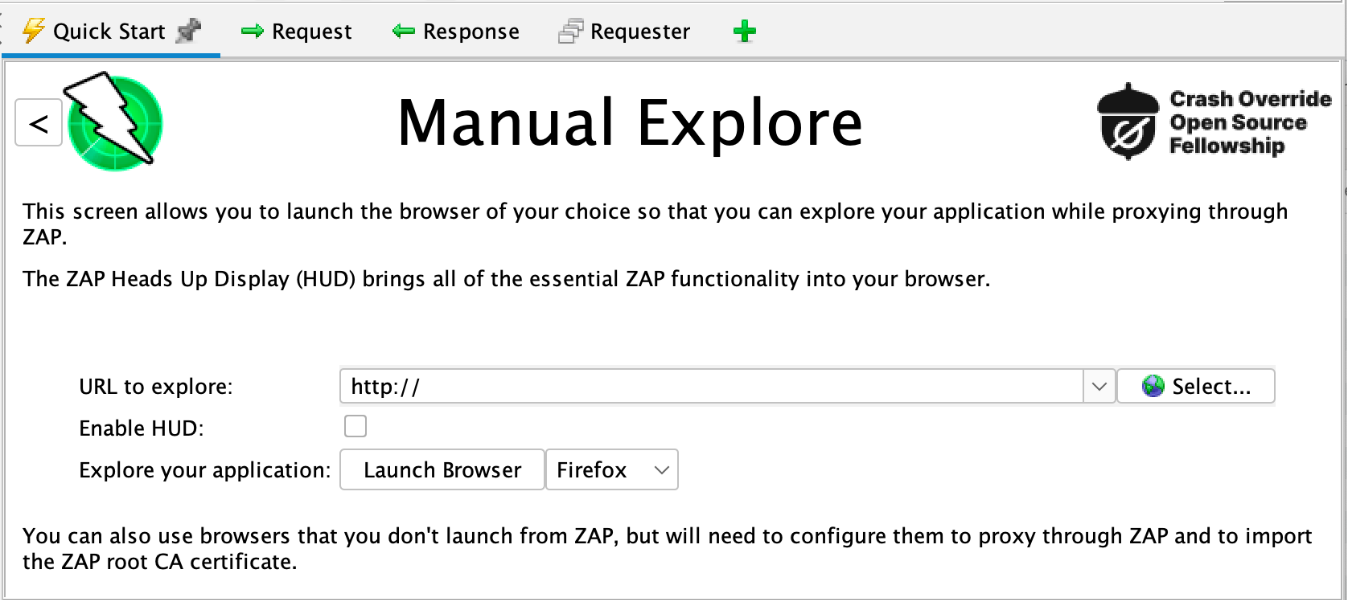
1. **Performans Sorunları**: Büyük ve karmaşık web uygulamalarında performans sorunları yaşanabilir.
2. **Öğrenme Eğrisi**: Yeni başlayanlar için bazı özelliklerin ve kullanımın öğrenilmesi zaman alabilir.
3. **Aktif Tarama Riskleri**: Aktif taramalar, test edilen web uygulamasına zarar verebilir veya uygulamanın performansını etkileyebilir.
4. **Yanlış Pozitifler**: Bazen yanlış pozitif sonuçlar verebilir, bu da güvenlik uzmanlarının manuel olarak sonuçları doğrulamasını gerektirir.
5. **Sınırlı Mobil Desteği**: Mobil uygulamaların test edilmesinde sınırlı yeteneklere sahip olabilir.

OWASP ZAP, web uygulamalarının güvenlik testlerini yapmak için güçlü ve esnek bir araçtır. Ancak, tüm güvenlik testlerinde olduğu gibi, sonuçları dikkatle değerlendirip manuel doğrulama yapmak önemlidir.

[**https://github.com/zaproxy**](https://github.com/zaproxy)github dokümantasyonu

<https://github.com/iriusrisk/zap-webdriver> tutorial

<https://www.zaproxy.org/getting-started/#overview> Sitesi



**2.SqlMap**

Açık kaynaklı bir araç olan SQLMap, veri tabanlarında SQL Injection güvenlik açıklarını otomatik olarak tespit edip istismar etmeye yarar. SQLMap, çeşitli veri tabanı yönetim sistemlerini (DBMS) destekler ve veritabanı yapılarını çıkarmak, veritabanı kullanıcı bilgilerini ele geçirmek gibi birçok gelişmiş özellik sunar.

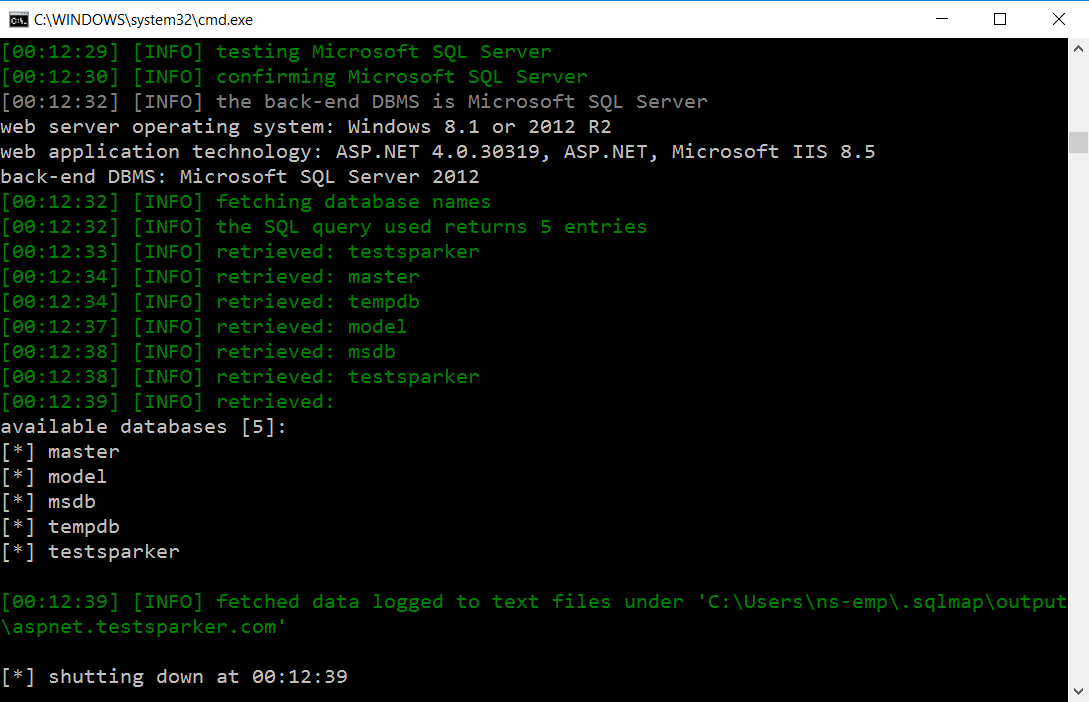
## **Basit Kullanım Örnekleri**

İlk olarak, bazı basit parametrelerin kullanımına dair örneklerle başlayalım. Bilindiği üzere en yaygın kullanılan sqlmap parametresi URL’i ifade eden -u ya da --url parametresidir. Aşağıdaki kullanımda da görüldüğü üzere sqlmap’e hedef sitenin URL’ini tanıtır:

[](https://www.netsparker.com.tr/blog/web-guvenligi/ileri-seviye-sqlmap-kullanimi/)

Bu komut sonucunda, eğer taratılmasını istediğimiz URL’de SQL Injection zafiyeti mevcutsa, zafiyetin türü, hangi DBMS (Database Management System)’in kullanıldığı, web uygulama dili ve sunucu versiyonu gibi bazı bilgilere ulaşmamız mümkün olacaktır. Devamında SQL Injection’ın var olduğunu öğrendiğimiz URL vasıtasıyla saldırıya --dbs parametresini ekleyerek mevcut veri tabanlarının listesini ekrana getirebiliriz:

sqlmap -u “http://aspnet.testsparker.com/Products.aspx?pId=4” --dbs

[](https://www.netsparker.com.tr/blog/web-guvenligi/ileri-seviye-sqlmap-kullanimi/)

Ekran görüntüsünde de görüldüğü üzere mevcut veri tabanlarının listesini elde etmiş olduk. Bundan sonra, sırayla tabloları, kolonları ve kayıtları aşağıdaki parametreleri kullanmamız gerekiyor:

sqlmap -u “http://aspnet.testsparker.com/Products.aspx?pId=4” -D testsparker --tables

sqlmap -u “http://aspnet.testsparker.com/Products.aspx?pId=4” -D testsparker -T tablo --columns

sqlmap -u “http://aspnet.testsparker.com/Products.aspx?pId=4” -D testsparker -T tablo -C kullanici,parola --dump

**-D:** Bu parametre ile veri tabanı adını belirtmiş olduk.  
**--tables:** Belirttiğimiz veri tabanındaki tablo isimlerini getirmek için bu parametre kullanılır.  
**-T:** Tablo isimleri de alındıktan sonra kolonlarına ulaşılması istenen tablo adı bu parametre ile birlikte yazılır.  
**--columns:** Adı belirtilen tablodaki kolonlar getirilir.  
**-C:** İstenilen kolon adı yazılır.  
**--dump:** Kolonlarda yer alan veriler ekrana yansıtılır.

**Tüyo:** Tek tire (-) veya çift tire (--) kullanıldığında parametre ile parametrenin aldığı değer arasında boşluk ya da eşittir işaretinden (=) hangisinin kullanılması gerektiğini karıştırıyorsanız tek tireli parametrelerde boşluğun, çift tireli parametrelerde ise değer girilmeden önce eşittir işaretinin kullanıldığını aklınızda tutabilirsiniz.

**Not:** Herhangi bir database’den veri çekebilmek için sqlmap ile test ettiğimiz uygulamanın veri tabanı kullanıcısının gereken yetkileri haiz olması zorunluluğunu unutmamak gerekir.

Bu şekilde bir kullanımı vardır daha iyi anlayabilmek için:

<https://github.com/sqlmapproject/sqlmap>

<https://www.netsparker.com.tr/blog/web-guvenligi/ileri-seviye-sqlmap-kullanimi/>

<https://sqlmap.org/>

**3.Nessus**

Ticari bir üründür, geniş bir güvenlik açığı veri tabanına sahiptir, ağ taramaları yapar, çeşitli işletim sistemleri ve uygulamaları destekler. Dünyada birçok kullanıcısı bulunan güvenlik zafiyeti tarama programıdır.  Nessus Professional, Nessus Manager, Nessus Home ve Nessus Cloud sürümleri mevcuttur. Fiziksel, sanal ve bulut ortamlarında güvenlik zafiyetlerinin ve zararlı yazılımların tespitini sağlar.

Nessus sayesinde kurumun saldırı düzeyleri küçültülür.Nessus, yüksek-hızlı varlık tespiti, yapılandırma denetimi, hedef ayrımlama, zararlı yazılım tespiti ve hassas veri tespiti özelliklerinin yanı sıra pek çok başka özellikle birlikte sunulmaktadır.Geniş ve kapsamlı bir tarama sağlar.Saldırıları sınıflandırma özelliği ile ölçeklendirilebilir bir yapı sunar.Düşük maliyet sunar.Her geçen gün gelişen saldırı teknik ve taktiklere karşı güncel çözümler ve tarama teknikleri sunmaktadır.Nessus Home aracı ile ücretsiz hizmet sağlamaktadır.Tek kullanıcılı lisan veren temel ihtiyaçları sağlayan Nessus Proffesional, üst düzey ihtiyaç için Nessus Manager veya Nessus Cloud’a sürümlerini sunmaktadır.

[**https://www.tenable.com/products/nessus**](https://www.tenable.com/products/nessus)

[**https://laccart.medium.com/nessus-nedir-ve-nas%C4%B1l-kurulur-27435d427f0b**](https://laccart.medium.com/nessus-nedir-ve-nas%C4%B1l-kurulur-27435d427f0b)

****

**11.Duman Testi**

Duman testi (Smoke Testing), yazılım geliştirme sürecinde yapılan ilk test türlerinden biridir ve temel işlevlerin düzgün çalışıp çalışmadığını kontrol eder. Bu test, yazılımın daha kapsamlı testlere hazır olup olmadığını belirlemek için kullanılır. Duman testi, adını elektronik donanımın ilk testi sırasında, cihazın duman çıkarıp çıkarmadığını kontrol etmekten alır. Yazılım dünyasında ise, bu test yazılımın "yanlış" bir durumda olup olmadığını kontrol eder.

**1.Selenium**

 **Çapraz Tarayıcı Desteği**:

* Selenium, Chrome, Firefox, Safari, Internet Explorer ve Edge gibi birçok tarayıcıda çalışabilir.

 **Programlama Dili Desteği**:

* Java, C#, Python, Ruby, JavaScript ve PHP gibi çeşitli programlama dillerini destekler.

 **Açık Kaynak**:

* Selenium, açık kaynaklı bir araçtır ve ücretsiz olarak kullanılabilir.

 **WebDriver API**:

* WebDriver API, doğrudan tarayıcılarla iletişim kurar, bu da daha hızlı ve güvenilir testler sağlar.

 **Paralel Test Koşumu**:

* Selenium Grid ile testleri paralel olarak çalıştırarak test sürelerini kısaltabilirsiniz.

 **Büyük Topluluk ve Destek**:

* Geniş bir kullanıcı ve geliştirici topluluğu sayesinde çok sayıda kaynak, eğitim materyali ve forum desteği bulunur.

 **Entegrasyon**:

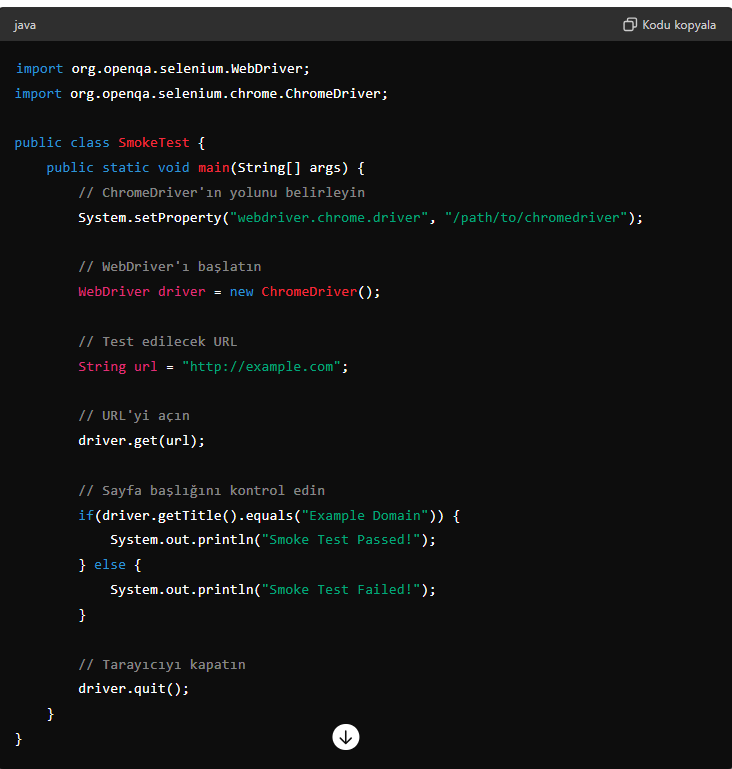
* Jenkins, Maven, TestNG ve JUnit gibi birçok CI/CD aracı ve test çerçevesi ile entegrasyon sağlar.

**Mobil Desteğin Sınırlı Olması**:

* Selenium, web uygulamaları için idealdir ancak mobil uygulamalar için Appium gibi ek araçlar gerektirir.

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

<https://www.youtube.com/watch?v=g6gQtN0H-sE> çok güzel bir örnek yapıyor



**2.**[**PhantomJS**](https://phantomjs.org/)

PhantomJS, web uygulamalarının otomatik test edilmesi ve ağ izleme işlemlerinde kullanılan, tarayıcı motorunu arka planda çalıştıran bir "headless" web tarayıcısıdır.

 **Headless Tarayıcı**:

* PhantomJS, grafiksel bir kullanıcı arayüzü olmadan çalışan bir web tarayıcısıdır. Bu, hızlı ve kaynak açısından verimli testler sağlar.

 **WebKit Tabanlı**:

* WebKit tarayıcı motorunu kullanır, bu da Safari ve eski Chrome tarayıcılarıyla uyumluluğu artırır.

 **Hızlı Yürütme**:

* Grafik arayüzü olmaması nedeniyle, testler genellikle tam tarayıcılara göre daha hızlı çalışır.

 **JavaScript API**:

* Web sayfalarıyla etkileşim kurmak için zengin bir JavaScript API'si sunar.

 **Ağ İzleme**:

* Ağ trafiğini izleme ve analiz etme yeteneği sunar. Bu, performans testi ve hata ayıklama için yararlıdır.

 **Ekran Görüntüsü Alma**:

* Web sayfalarının ekran görüntülerini alabilir, bu da görsel regresyon testleri için faydalıdır.

### Avantajlar

1. **Performans**:
   * Headless olması nedeniyle testler hızlı ve kaynak tüketimi düşüktür.
2. **Kolay Kurulum ve Kullanım**:
   * Kurulumu ve kullanımı nispeten kolaydır. Node.js ve diğer dillerle entegrasyonu kolaydır.
3. **Entegre Edilebilirlik**:
   * Selenium gibi test otomasyon çerçeveleriyle kolayca entegre edilebilir.
4. **Ağ İzleme ve Ekran Görüntüsü Alma**:
   * Bu özellikler, testlerin daha kapsamlı ve görsel doğrulama için kullanılmasını sağlar.
5. **Script Yazma Yeteneği**:
   * PhantomJS, doğrudan JavaScript kullanarak test senaryoları yazmayı mümkün kılar.

### Dezavantajlar

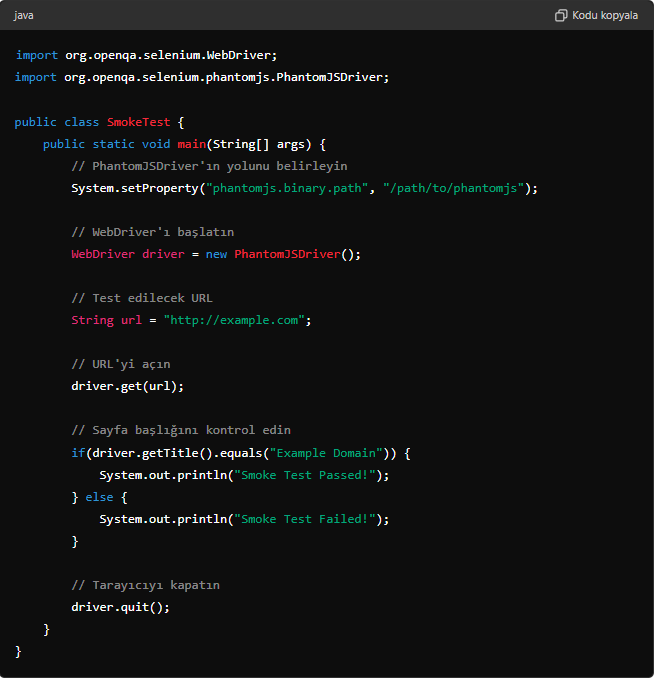
1. **Destek ve Geliştirme**:
   * PhantomJS, artık aktif olarak geliştirilmiyor ve bazı modern web teknolojilerini tam olarak desteklemeyebilir.
2. **Tarayıcı Uyumluluğu**:
   * WebKit tabanlı olduğundan, tarayıcı uyumluluğunu tam olarak test edemeyebilir. Özellikle, modern tarayıcıların özelliklerini desteklemekte yetersiz kalabilir.
3. **Hata Ayıklama**:
   * Grafik arayüzü olmadığı için, hata ayıklama ve görsel doğrulama zor olabilir.
4. **Güvenilirlik**:
   * Bazı karmaşık uygulamalar için tam olarak güvenilir olmayabilir.

PhantomJS, genellikle Selenium veya CasperJS gibi araçlarla birlikte kullanılır

[**https://phantomjs.org/**](https://phantomjs.org/)

[**https://github.com/jamesshore/ll13\_phantomjs\_smoke\_test**](https://github.com/jamesshore/ll13_phantomjs_smoke_test)

[**https://www.letscodejavascript.com/v3/comments/lessons\_learned/13**](https://www.letscodejavascript.com/v3/comments/lessons_learned/13) **güzel bir tutorial uygulaması olmuş**

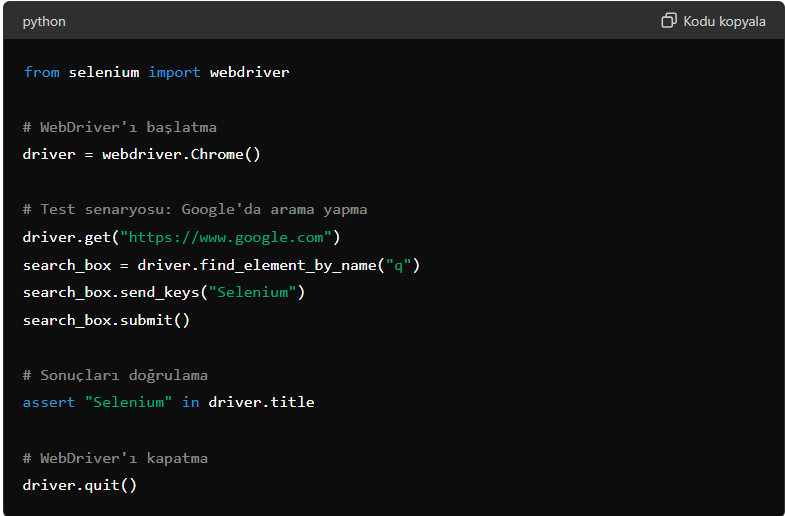
****

**12.Regresyon Testi**

**1.Selenium**

### Selenium ile Regresyon Testi Nasıl Yapılır?

1. **Test Senaryolarını Belirleme:**
   * Regresyon testi için otomatikleştirilmiş test senaryoları belirlenir. Bu senaryolar, yeni kod değişikliklerinin mevcut işlevselliği bozmadığını doğrulamak için kullanılır.
2. **Selenium WebDriver Kullanarak Testleri Yazma:**
   * Test senaryolarını otomatikleştirmek için Selenium WebDriver kullanılır. Örneğin, Python'da Selenium WebDriver ile bir test senaryosu şöyle yazılabilir:



 **Testleri Çalıştırma ve Raporlama:**

* Test senaryoları çalıştırılır ve sonuçlar raporlanır. Hatalar ve başarısızlıklar analiz edilerek düzeltmeler yapılır.

 **Sürekli Entegrasyon (CI) ile Entegrasyon:**

* Jenkins, Travis CI, CircleCI gibi CI/CD araçlarıyla entegre edilerek, her kod değişikliği sonrasında otomatik olarak regresyon testleri çalıştırılır ve raporlanır.

### Selenium Kullanmanın Avantajları

* **Çapraz Tarayıcı Desteği:** Selenium, çeşitli tarayıcılarda çalışır ve çapraz tarayıcı testlerini destekler.
* **Esneklik:** Farklı programlama dillerini destekler, bu da geliştiricilerin tercih ettikleri dilde test yazmalarını sağlar.
* **Büyük Topluluk ve Destek:** Geniş bir kullanıcı topluluğu ve bol miktarda dokümantasyon ve kaynak bulunmaktadır.
* **Entegrasyon:** Selenium, CI/CD araçları ve diğer test otomasyon araçlarıyla kolayca entegre edilebilir.

### Selenium Kullanmanın Dezavantajları

* **Bakım Zorluğu:** Test senaryolarının bakımı, özellikle web uygulamaları sık sık değişiyorsa, zor olabilir.
* **Öğrenme Eğrisi:** Selenium'un tüm özelliklerini ve yeteneklerini öğrenmek zaman alabilir.
* **Performans:** Selenium testleri, manuel testlerden daha hızlı olsa da, büyük ve karmaşık test senaryolarında performans sorunları yaşanabilir.

<https://www.browserstack.com/guide/regression-testing-with-selenium> güzel bir örnek

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

**2.TestComplate**

TestComplete, farklı platformlarda çalışan uygulamaların kullanıcı arayüzünü otomatik olarak test edebilen bir test otomasyon aracıdır. Bu araç, hem script tabanlı hem de script'siz test otomasyonunu destekler, bu da hem teknik hem de teknik olmayan kullanıcıların testler oluşturmasını sağlar.

### TestComplete ile Regresyon Testi Nasıl Yapılır?

#### 1. **Test Projesi Oluşturma:**

* TestComplete ile ilk adım, bir test projesi oluşturmaktır. TestComplete, farklı türdeki projeleri destekler (web, masaüstü, mobil vb.).

#### 2. **Test Senaryolarını Kaydetme:**

* TestComplete, kullanıcı etkileşimlerini kaydedebilir. Bu, bir uygulama üzerinde yapılan işlemleri kaydederek otomatik test senaryoları oluşturmanızı sağlar.
* Kayıt işlemi sırasında, uygulama üzerinde tıklamalar, veri girişi, menü navigasyonu gibi işlemler kaydedilir.

#### 3. **Test Senaryolarını Düzenleme ve Geliştirme:**

* Kaydedilen test senaryoları düzenlenebilir ve geliştirilebilir. TestComplete, testlerin daha dinamik ve esnek olmasını sağlamak için çeşitli programlama dillerini (JavaScript, Python, VBScript vb.) destekler.
* Örneğin, koşullu ifadeler, döngüler, değişkenler gibi yapıları kullanarak test senaryolarını özelleştirebilirsiniz.

#### 4. **Testlerin Çalıştırılması:**

* Hazırlanan test senaryoları, tek bir tıklama ile çalıştırılabilir. TestComplete, farklı tarayıcılar ve cihazlar üzerinde testlerin çalıştırılmasını destekler.
* Testler, manuel olarak veya zamanlanmış görevler olarak çalıştırılabilir.

#### 5. **Test Sonuçlarını İnceleme ve Raporlama:**

* TestComplete, testlerin sonuçlarını ayrıntılı raporlarla sunar. Başarılı ve başarısız test adımları, ekran görüntüleri ve hata mesajları gibi bilgiler içeren raporlar oluşturur.
* Bu raporlar, testlerin kalitesini değerlendirmek ve olası hataları tespit etmek için kullanılır.

#### 6. **Sürekli Entegrasyon (CI) ile Entegrasyon:**

* TestComplete, Jenkins, TeamCity ve diğer CI/CD araçları ile entegre edilebilir. Bu entegrasyon sayesinde, her kod değişikliğinde otomatik olarak regresyon testleri çalıştırılabilir.
* CI entegrasyonu, testlerin düzenli ve otomatik olarak çalıştırılmasını sağlayarak yazılım kalitesini artırır.

### TestComplete Kullanmanın Avantajları

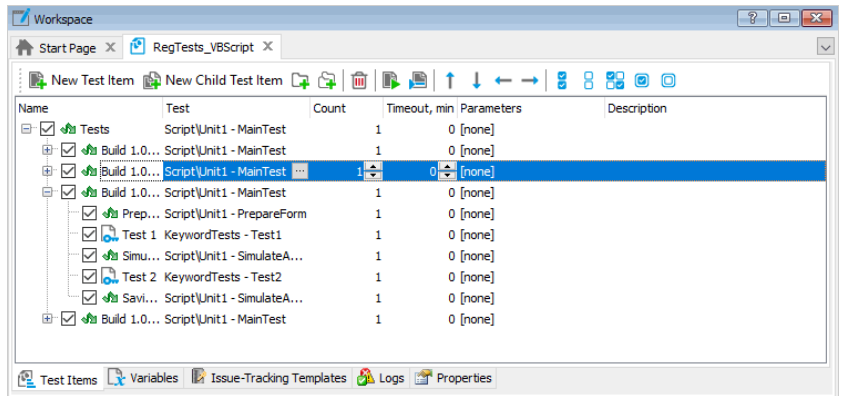
* **Kapsamlı Platform Desteği:** TestComplete, web, masaüstü ve mobil uygulamaların testlerini destekler.
* **Script ve Script'siz Test:** Hem kod yazarak hem de kaydedici kullanarak test oluşturma imkanı sunar.
* **Gelişmiş Raporlama:** Test sonuçlarını ayrıntılı ve görsel raporlarla sunar.
* **Kolay Entegrasyon:** CI/CD araçları ile kolayca entegre edilebilir.
* **Kullanıcı Dostu Arayüz:** Teknik olmayan kullanıcılar için bile kolay anlaşılır bir arayüze sahiptir.

### TestComplete Kullanmanın Dezavantajları

* **Maliyet:** TestComplete, ticari bir üründür ve lisans maliyetleri yüksek olabilir.
* **Öğrenme Eğrisi:** Gelişmiş özellikleri ve esnekliği nedeniyle öğrenme süreci biraz zaman alabilir.
* **Kaynak Tüketimi:** Bazı büyük ve karmaşık projelerde yüksek sistem kaynakları tüketebilir.

<https://support.smartbear.com/testcomplete/docs/testing-approaches/regression-testing.html>

kendi sitelerinde açıklanmış



3.**Ranorex**

Ranorex, otomasyon testi için kullanılan bir yazılım test otomasyon aracıdır. Özellikle GUI (Grafiksel Kullanıcı Arayüzü) tabanlı testlerde kullanılan Ranorex, farklı platformlarda (web, masaüstü, mobil) çalışan uygulamaların testlerini otomatikleştirmeyi sağlar. Ranorex, özellikle regresyon testleri gibi tekrarlayan ve karmaşık test senaryolarının yönetimini kolaylaştırmak için tasarlanmıştır.

### Ranorex Özellikleri ve Kullanım Alanları

1. **GUI Otomasyonu:**
   * Ranorex, farklı GUI tabanlı uygulamaların testlerini otomatikleştirmek için geliştirilmiştir. Web, masaüstü ve mobil uygulamalar üzerinde GUI bileşenlerine erişim sağlar ve bu bileşenler üzerinde test senaryolarını kolayca yürütebilir.
2. **Script Desteği:**
   * Ranorex, test senaryolarını oluşturmak için C# veya VB.NET gibi programlama dillerini destekler. Bu sayede geliştiriciler, testlerini daha dinamik hale getirebilir ve özelleştirebilir.
3. **Kayıt ve Oynat (Record and Playback):**
   * Ranorex, kayıt ve oynat özelliği sunar. Bu özellik sayesinde, kullanıcılar manuel olarak uygulama üzerinde yapacakları işlemleri kaydedebilir ve bu kayıtları temel alarak test senaryolarını otomatikleştirebilir.
4. **Gelişmiş Element Tanımlama ve Erişim:**
   * Ranorex, uygulama üzerindeki GUI bileşenlerine (örneğin, düğmeler, metin kutuları, menüler vb.) erişim sağlamak için gelişmiş element tanımlama yöntemleri sunar. Bu sayede test senaryoları daha güvenilir hale gelir.
5. **Raporlama ve Analiz:**
   * Ranorex, test sonuçlarını detaylı raporlar halinde sunar. Başarılı ve başarısız test adımları, ekran görüntüleri ve hata mesajları gibi bilgileri içeren raporlar, testlerin kalitesini değerlendirmek ve hataları tespit etmek için kullanılır.
6. **Entegrasyon Yeteneği:**
   * Ranorex, CI/CD araçları (Jenkins, TeamCity, Bamboo vb.) ile entegre edilebilir. Bu entegrasyon sayesinde, her kod değişikliğinde otomatik olarak testlerin çalıştırılması ve sonuçların raporlanması mümkün olur.
7. **Eğitim ve Destek:**
   * Ranorex, kullanıcılarına geniş bir eğitim ve destek ağı sunar. Kullanıcılar, ürünü etkin bir şekilde kullanabilmek için eğitim materyallerine ve topluluk desteğine erişebilirler.

### Ranorex Kullanmanın Avantajları

* **GUI Otomasyonu İçin Uygun:** GUI tabanlı uygulamaların otomasyonu için güçlü bir araçtır.
* **Kapsamlı Platform Desteği:** Web, masaüstü ve mobil uygulamalar üzerinde geniş kapsamda kullanılabilir.
* **Kullanıcı Dostu Arayüz:** Kayıt ve oynat gibi kullanıcı dostu özellikler sunar, teknik olmayan kullanıcılar için idealdir.
* **Gelişmiş Raporlama:** Detaylı raporlar ile test sonuçlarını analiz etme imkanı sunar.
* **Script Desteği:** C# veya VB.NET gibi programlama dillerini destekleyerek esneklik sağlar.

### Ranorex Kullanmanın Dezavantajları

* **Maliyet:** Ranorex, ticari bir üründür ve lisans maliyetleri yüksek olabilir.
* **Öğrenme Eğrisi:** Gelişmiş özellikleri nedeniyle öğrenme süreci biraz zaman alabilir.
* **Kaynak Tüketimi:** Bazı büyük ve karmaşık projelerde yüksek sistem kaynakları tüketebilir.

<https://support.ranorex.com/>

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

