



in/poyrazavsever



END-TO-END

End-to-End Test Mantığı



in/poyrazavsever



End-to-End Test Nedir?

End-to-End testler, bir uygulamanın tüm akışını gerçek bir kullanıcı gibi test etmeyi amaçlayan otomasyon sürecidir.

Bu testlerde sistemin tamamı, UI, API, veritabanı, yönlendirmeler ve kullanıcı etkileşimleriyle birlikte değerlendirilir.

Amaç, uygulamanın uçtan uca sorunsuz çalıştığını doğrulamaktır.



in/poyrazavsever



E2E Testlerin Sağladığı Avantajlar

E2E testler, uygulamada iş akışı bozulduğunda bunu erken aşamada fark etmenizi sağlar.

Özellikle frontend projelerinde küçük bir UI değişikliği bile kritik bir kullanıcı yolunu bozabilir.

E2E testler, bu kırılmaları hızlı şekilde ortaya çıkarır ve üretim ortamına hatalı kod çıkma riskini azaltır.



in/poyrazavsever



E2E Testler Diğer Test Türlerinden Nasıl Ayrılır?

Yazılım testleri genellikle üç seviyede ele alınır:

- 1. Unit test**
- 2. Integration test**
- 3. End-to-End test**

E2E, en üst seviyedir ve sistemin bütününe test eder.

Unit testler tek bir fonksiyonu, integration testler ise iki birimin birlikte çalışmasını test eder.



in/poyrazavsever



Bir E2E Test Senaryosu Nasıl Oluşturulur?

Bir E2E test senaryosu, gerçek bir kullanıcının uygulamada izlediği adımların teknik olarak modellenmiş halidir.

Genellikle şu yapıda hazırlanır:

- 1. Başlangıç durumu**
- 2. Kullanıcı aksiyonları**
- 3. Beklenen sonuç**

Bu yapı, okunabilir ve sürdürülebilir testler üretmeyi kolaylaştırır.



in/poyrazavsever



Basit Bir Kullanıcı Senaryosu

Aşağıdaki akış basit bir giriş senaryosudur:

- **Giriş sayfasına git**
- **Kullanıcı bilgilerini gir**
- **Giriş butonuna bas**
- **Kullanıcı panelinin açıldığını doğrula**

Bu akış gerçek bir kullanıcıyı temsil eder.



in/poyrazavsever



Cypress Kullanarak E2E Testi

Cypress, modern frontend projelerinde en çok tercih edilen E2E test aracıdır.

Basit bir giriş testi aşağıdaki gibidir.

```
describe('Login Test', () => {  
  it('should login successfully', () => {  
    cy.visit('/login');  
    cy.get('#email').type('test@example.com');  
    cy.get('#password').type('123456');  
    cy.get('button[type=submit]').click();  
    cy.url().should('include', '/dashboard');  
  });  
});
```



in/poyrazavsever



Okunabilir Test Yazma İlkeleri

E2E testlerin sürdürülebilir olması için okunabilirlik kritik öneme sahiptir.

Test adımları kısa, düzenli ve tek bir amacı karşılayan parçalardan oluşmalıdır.

Testler, uygulamanın işleyişini anlamayan biri tarafından bile okunabilir olmalıdır.



in/poyrazavsever



Page Object Pattern Nedir?

Page Object Pattern, E2E testlerde kullanılan popüler bir yapıdır.

Sayfalar veya bileşenler için tekrar kullanılabilir fonksiyonlar tanımlanır.

Bu yaklaşım, testlerin tekrar eden kodlardan arınmasını sağlar.



in/poyrazavsever



Page Object Pattern Uygulaması

Bu yapı sayesinde test dosyaları sadeleşir.



```
class LoginPage {  
  visit() { cy.visit('/login'); }  
  typeEmail(value) { cy.get('#email').type(value); }  
  typePassword(value) { cy.get('#password').type(value); }  
  submit() { cy.get('button[type=submit]').click(); }  
}
```

```
export default new LoginPage();
```



in/poyrazavsever



Test Ortamı Yapılandırması

E2E testler genellikle üretim ortamından ayrı, kontrollü bir test ortamında çalıştırılır.

Bu ortamda test kullanıcıları, test veritabanı ve geliştirme API adresleri bulunur.

Amaç testlerin izole bir alanda stabil şekilde çalışmasını sağlamaktır.



in/poyrazavsever



Senkronizasyon Problemleri

E2E testlerde hataların bir kısmı zamanlama kaynaklıdır.

Arayüzde bir eleman gecikmeli yüklenebilir.

Cypress gibi araçlar bu beklemelemleri otomatik yönetir,
ancak doğru seçicileri kullanmak test stabilitesini artırır.



in/poyrazavsever



Testleri Pipeline'a Dahil Etmek

E2E testler genellikle CI/CD süreçlerinin bir adımına eklenir.

Her deploy öncesi otomatik testler çalıştırılır ve kritik hatalar tespit edilirse süreç durdurulur.



in/poyrazavsever



End-to-End Testlerin Sağladığı Güvence

E2E testler, uygulamanın gerçek kullanıcı akışında sorunsuz şekilde çalıştığını doğrulayan en güçlü test biçimidir.

Doğru senaryolar, düzenli bir test seti ve okunabilir bir yapı ile projelerde yüksek güvenlik ve süreklilik sağlanır.



Kendi Senaryonu Oluřturmaya Bařla
Öğrendiklerini uygulamak için basit bir
kullanıcı yolunu seç ve adımlarını yaz!



in/poyrazavsever



@poyraz_avsever



@poyrazavsever