

**KOCAELİ ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**YAZILIM LABORATUVARI - II PROJE - II**

**Mobil Sorgular**

Açıklama: Şekil**ENGİN YENİCE**

**190201133**

**CEMRE CAN KAYA**

**190201137**

**KOCAELİ 2020**

Mobil Sorgu Uygulaması

Cemre Can Kaya   
Bilgisayar Mühendisliği  
Kocaeli Üniversitesi  
190201137

Engin Yenice  
Bilgisayar Mühendisliği  
Kocaeli Üniversitesi  
190201133

*Özet*— todo

Anahtar Kelimeler— todo

# Giriş

**todo**

# Proje Mimarisi

Projenin arka planı (backend) kurumsal mimariye uygun bir şekilde geliştirilmiştir. Proje yapısı gereği 5 temel katmana parçalanmıştır.

Projenin ön planı (front-end) parçala yönet mantığı ile geliştirilmiş olup her bir işlem parçacığı ayrı bir bileşende (component) üzerinde yapılmaya çalışılmıştır.

## Arka Plan (Backend) Yapısı

Proje yapısı gereği 5 temel katmana parçalanmıştır. Bu katmanların detayları bu başlık altında açıklanmıştır.

### Core Katmanı

Bu katman proje içerisinde bulunması gereken temel bileşenleri bulundurmaktadır.

#### **Entities:**

Oluşturduğumuz nesnelerin daha somut ve yönetilebilmesi için temel arayüzleri (interface) bu klasör içerisinde tutulmaktadır.

* **IEntity:** Temel nesnelerimizin temel arayüz (interface) sınıfıdır.
* **IDto:** Uzun ismi **Veri iletim nesnesi (Data transfer object)** olarak geçmektedir. Temel nesnelerimizden Kullanıcı arayüzüne (UI (User Interface (Front-end))) göndermek istediğimiz nesne tanımlamalarının temel arayüz (interface) sınıfıdır.

#### **Utilities:**

Projenin genelinde kullanabileceğimiz araçlar bu klasör altında tutulmaktadır.

* **Results:** İş (Business) katmanında kullanacağımız metotların geriye dönüş değerlerinin daha yönetilebilir ve daha düzenli olması için oluşturduğumuz bir araç sınıfıdır. Bu sınıfı temel olarak özetlenecek olursak. İçerisinde temel olarak 2 adet değişken bulunmaktadır. Bu değişkenler mesaj ve başarı (message, success) durumu olarak isimlendirilmektedir. Veri gönderilmesi durumunda miras verdiği alt sınıfta ise data (veri) değişkeninin bulunduğu ayrı bir dönüş tipi bulunmaktadır. Sınıf içerisinde ki değişkenlerin daha kolay yönetilmesi için başarı (success) durumuna göre alt sınıflar oluşturulmuştur. Bu sınıfların çağrılması durumunda başarı durumu otomatik olarak belirlenmektedir.

### Entities Katmanı

Entity ve Dto arayüzlerinden (interface) örnek alan veri sınıfları bulunmaktadır.

#### **Concrete**

Entity arayüzünden (interface) örnek alınan veri sınıfları bulunmaktadır.

#### **Dto:**

Dto arayüzünden (interface) örnek alınan veri sınıfları bulunmaktadır.

### DataAccess Katmanı

Veri havuzunun yönetilmesinden görevli katmandır. Bu katmanımızı ileride geliştirmeye açık olması için arayüzler (interface) kullanarak geliştirdik. Bu sayede ileride bir gerektiğinde başka yapılara geçmeyi planladık. Şuanda veri havuzumuzu bellekte (In Memory) olarak tutuyoruz.

#### Abstract

Veri sınıflarımızın arayüzlerinin tutulduğu klasör.

#### Concrete

Veri sınıflarımızın tutulduğu klasör.

### Business Katmanı

Projenin iş kodlarının yazıldığı katmandır. Bu gerekli işlemlerin yönetildiği katmandır. Temel olarak 4 klasöre bölünmüştür.

#### Abstract

Business sınıflarımızın arayüzlerinin tutulduğu klasör.

#### Concrete

Business sınıflarımızın tutulduğu klasör.

* **IndexerManager:** WebAPI tarafından gönderilen isteklerin gerekli helper sınıflarına yönlendirilmesini sağlayan ve dönen sonuçların geri WebAPI tarafına yönlendiren sınıftır.

#### DependencyResolvers

Bağımlılıkların çözülmesi ve isim havuzuna aktarılması için kullanılmaktadır. (Paket olarak Autofac kullanılmıştır.)

#### Helpers

Projenin akışını sürdürecek helper sınıfları yazılmıştır. Helper sınıflarımızın arayüzlerinin tutulduğu abstract klasörü ve örnek alındığı concrete klasörü bulunmaktadır.

### WebAPI katmanı

Ön Plan (Front-end) kısmından gelen istekleri karşılayıp gerekli dönüşleri yapmakla görevli olan katmandır.

## Ön Plan (Frontend) Yapısı

Projenin kullanıcı ara yüzü (User Interface) ekranıdır. Arka plan (Backend) tarafında yazılan kodların kullanıcıya görüntüsel olarak aktarıldığı taraftır.

### Ana Bileşen (Component) (app.component)

Projenin çalıştığı ana bileşendir. Bu bileşen içerisinde sabit ve dinamik bileşenler çağrılmaktadır.

Sabit bileşenler direk ana bileşen içerisinden çağrılmaktadır. Dinamik bileşenler ise rotalama (routing) yapısı ile ekrana getirilmektedir.

# Yontem

**todo**

# Nasıl Kullanılır

**todo**

# Çevrimiçi Linkler

## Backend (Arka Plan)

**todo**

## Frontend (Kullanıcı Arayüzü (UI))

**todo**

# Kaynakça

### <https://enginyenice.com/>

### <https://enginyenice.com/seo-ve-google-analytics/>