* Sanal bir şirket olan Akar yazılım ın tanıtım sayfası, admin paneli, iş başvuruları ve başvuru statülerini canlandırabileceğimiz bir ortam oluşturulmuştur.
* Temel seviyede bakıldığında ilgili database diagram şekilde gözüktüğü gibidir.

metin, ekran görüntüsü, diyagram, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Proje temeli yukarıdadır. Kullanıcı ve roller arasında çoğa çok bir ilişki mevcuttur.
* User tablosunda belirli kolonlar var. başvurular ile user lar arasında da çoka çok bir ilişki mevcuttur.
* **Sistemde 2 tip kullanıcım olacak bunlardan birisi Admin role ü için, diğerleri ise member rolü için geçerlidir.**
* **Member role ü olan user lar oluturulan iş ilanına başvurabilecektir.**
* **Basit bir iş başvuru sistemi yapacağız.**
* Proje katmanları ile oluşturulabilir bir vaziyettedir.
* İlk etapta bizler Common katmanı (Proje içerisinde Core katmanı olarak geçer) Costume Status Resultları dönebilmemiz için bir Result sınıfı ve DataResult sınıfı oluşturduk.
* **Çalışığımız şirket adı** + **Proje adı** şeklinde bir solution ismi oluşturuyorduk. **Dll** isimleri ise **şirket adı + proje adı + katman ismi** şeklinde oluşturacağız.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Response yapımızı artık Common katmanında ekleyeceğiz. Biz bu response yapımızı Common (shared – core içerisinde yapılandırıyorduk )
* Bir statü enum u, message, Fluent validasyon için ilgili list<ValidatonError> larımızı property olarak ekleyeceğiz. (success, validation Error, NotFound Statüs )

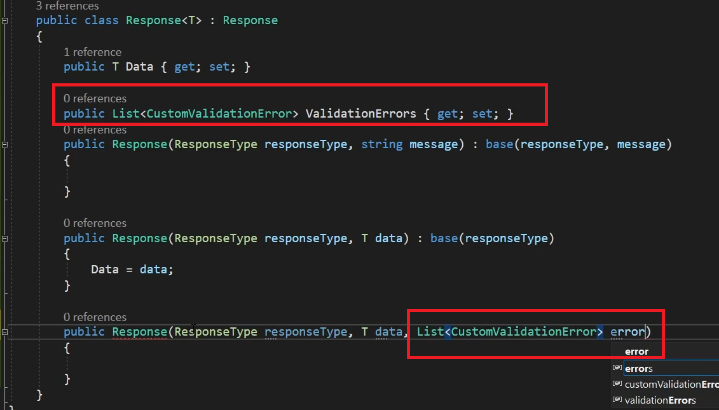
metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Buna uygun bir şekilde data Resultarınızı ve Result sınıflarınızı oluşturabilirsiniz. Örnekte herhangi bir response taşımayan bir yapı kurgulanmıştır.
* Data taşımayan response type ta bir validation errorları koymamış olabiliriz. Isteğe bağlı olarak opisyonel olarak konulabilir.
* Validation error için bir class ve bir Extension methoda ihtiyacımız var. Property name ve description için gereklidir.

metin, yazılım, multimedya yazılımı, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu



Business e aktarılan data business ten validasyon a takılırsa uı a geri gönderilmesi gerekecektir.

* Entity sınıfını ekleyelim bu aşamada.
* Bizim burada tüm entitylerimiz için id propertysi olmasını isteriz. Find propertysi için. Bu noktada tabi base entity oluşturarak ilgili tüm propertyleri burada tanımlar sonrasında tüm entitylerimizi buradan kalıtım almalıyız.
* Find methodu olacak günün sonunda o yüzden tüm entitylerimizde primary key olarak id bulunmadı geri kalanları manüel indexleyebilirsiniz.
* Bu işlem için Baseentity oluşturacağız.
* BaseEntity için tüm ortak olacak propertyler eklenebilir.

metin, çizgi, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Tüm entityler BaseEntity den kalıtım alacaktır.
* Tüm entityler için ilgili düzenlemeleri gerçekleştir.
* AdvertiesmentsUser çoka çok tablosu için bazı düzenlemeler yapılacaktır. Burada bazı properryler üzerinde durmamız gerekecektir.
* Ilgili userlar buraya başvuru yapabilmeleri için yapılmış olan bir tablodur.
* Bu tabloya atılan her bir kayıt ilgili kullanıcı için iş başvurusu manasına gelmektedir.
* Gender tablosunda neler olabilir? erkek, kadın ve belirtmek istemiyor olabilir. Bu tarz işlemlerde sadece içerisinde parameter için data barındıran tablolara bizler lookuptable diyoruz.

metin, diyagram, sayı, numara, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

* LookUpTable da propertye özgü bir iş yapılacaksa eğer (örnek vermek gerekirse tecilli olan asker adayları için ) ilgili tecil tarihi lookup ta değil operasyon tablosunda tutulabilir.

metin, diyagram, sayı, numara, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldu

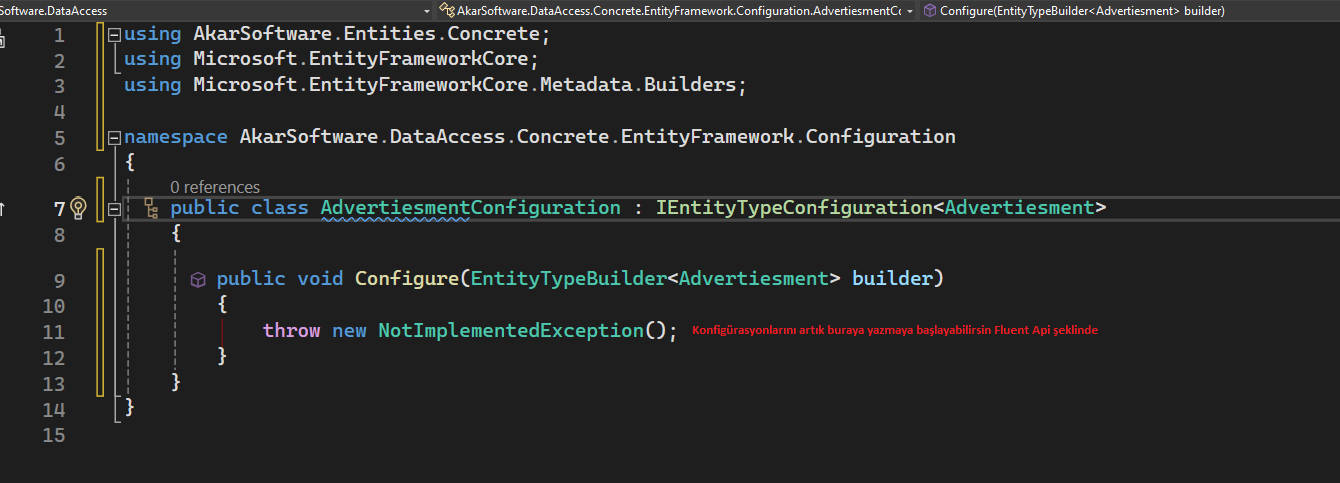
* Look up tablolarını eklemeye başlayalım.
* Entity Framework tarafında Desinger kısmın uı kısmına kurmamız gerecektir.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Lookuptable larda veritabanı yansımasına çoğul isim getirmeyebilirsiniz. LU ile isimlendirmenize başlayabilirsiniz.

* Configürasyonlarınızı Configurations isminde bir klasör içerisinde oluşturabilirsiniz.
* Aşağıdaki şekilde ilgili entity niz için bir configuration file hazırlayabilirsiniz.



* Bir entity e varsayılan değer atamasını hem fluent api vasıtası ile hemde property vasıtası ile gerçekleştirebilirsiniz(default değer ile ekleyebilirsin)

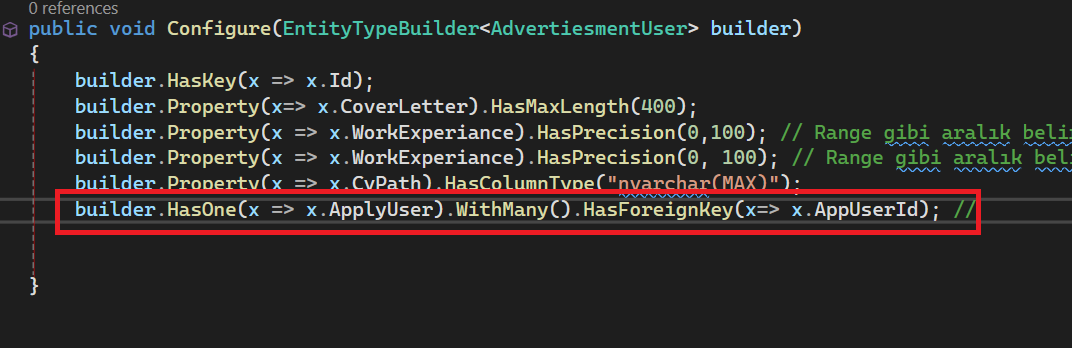
metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Ilişkleri belirtirken bir noktada belirtmeniz yeterlidir. Tek property olarak geçilen alanlarda foreign key belirtmeniz yeterlidir. Karışıklığı azaltamk için yoksa karşışıklıda belirleyebilirsiniz.
* Core katmanında ben Appuser içerisinde başvurulan işleri property olarak tutmadım. Bire çok ilişki kurabilmek için bir tarafa liste bir tarafa da tek property koyma zorunluluğum bulunmamaktadır. Fluent api içerisinde de ilgili ilişkide Withmany içerisinde ilgili alanı boş bırakabilirim.



metin, ekran görüntüsü, yazı tipi, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu