# **NOTLAR**

* Base Controller: **Base Controller ‘ı** oluşturmaktaki amaç projemiz içerisinde yer alan diğer **Controller’ın**içerisinde ortak olarak yer alan bazı propertyler ya da tanımlar olabilir. Bunları her defasında ayrı ayrı Controller içerisine yazmaktansa **Base** bir **Controller**oluşturarak bunun içerisinde ortak kullanılan nesneleri tanımlarız. Bu tanımlamaları yaptıktan sonra da diğer oluşturulan **Controller** ı oluşturduğumuz bu **BaseController**sınıfından miras alabiliriz.
* UnitOfWork sınıfı her controller için ortaktır. Çünkü, DbContext içerisinde yaptığımız her değişikliğin, veritabanına yansıması için, bu sınıf içerisindeki saveChanges metodunu kullanıyoruz.
* Solid prensipleri 5 önemli prensipten oluşur. Robert Martin tarafından öne sürülmüştür.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Solid prensiplerinin amacı kolay değiştirilebilir, sürdürülebilir sistemler inşaa edebilmektir.
* S- Single Responsibility: Karşılığı tek sorumluluktur. Tek sorumluluktan kasıt bir classınızın birden fazla iş kümesini yapmaması gerekiyor. Yani yaptığı işlere göre classlarımızı veya metotlarımızı konumlandırmamız lazım.
* Aşağıdaki görsel bir Single Responsibility’ e aykırıdır. Hem birproperty var hem validation işlemi var hem de bir metot tanımladık.
* Bunu düzeltmek için de hepsini bölmeliyiz. Her classımızın sadece bir sorumluluğu olmalı.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

* O- Open Closed: Nesneler değişime kapalı fakat genişlemeye açık olmalıdır. Yani classları oluşturduktan sonra onları birdaha değiştiremezsiniz.
* L- Liskov Substitution: Bu prensip bize der ki S eğer T’ nin bir alt türü ise S’ nin, T’ nin yerine geçebilmesi gerekir.
* I- Interface Segration: Alakalı interfaceleri parçalamalısın. Yani bir interface’ te alakasız bir şey olmamalı.
* D- Dependency Inversion: Bağımlılıkları tersine çevirmektir. Biz new kullandığımızda artık ona bağımlı hale geliriz. Bu bağımlılığı ortadan kaldırmak için ise Dependency Inversion’ dan yararlanıyoruz.
* Readonly kullanılırsa bir şey sadece constructor’ ın içerisinde setlenebilir. Yani bunu başka bir yerde setleyemezsiniz.

Örnek;

private readonly ICategoryRepository \_repository;

* Entity Framework Core açık kaynak kodlu ve platformdan bağımsızdır. Bir olaylam aracıdır. Veritabanındaki verilere erişmek veya bunları değiştirmek, eklemek, güncellemek, silmek gibi işleri kolaylaştıran bir mekanizma sağlar.
* Entity Framework Core ADO.NET geliştirmesidir.