**ASP.NET CORE 7.0 LOCALİZATİON (ÇOKLU DİL DESTEĞİ)**

ASP.NET Core, uygulamalarınızı birden fazla dilde destekleyebilmeniz için yerleşik olarak çoklu dil desteği ve lokalizasyon özellikleri sunar. Bu sayede, kullanıcılarınıza farklı dillerde içerik sunabilir ve uygulamanızın küresel olarak erişilebilir olmasını sağlayabilirsiniz. Bu yazımda sizlere localization’ı detaylı bir şekilde yapmayı öğreteceğim.

Lokalizasyonu etkinleştirmek için, ASP.NET Core projenizin Startup.cs dosyasında bazı yapılandırmalar yapmanız gerekmektedir.

İlk olarak program.cs dosyasına Localization’ı projemize dahil ediyoruz.



Sonrasında ise resources dosyalarımı nereye koyacağımızı belirtiyoruz.

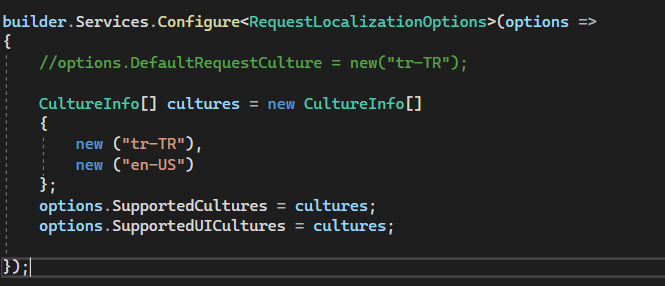


Resources adlı klasörümüzü projeye ekliyoruz.



Sonrasında program.cs dosyamızın içerisinde projemizde hangi dillerin destekleneceğini ayarlıyoruz.

//Yorum satırında ki kod, hiçbir işlem yapılmazsa projemiz başlangıçta türkçe gelsin karşılığıdır.



Bu işlemleri gerçekletirdikten sonra UseRequestLocalization çağrılarak etkinleştirilmelidir.



**CONTROLLER DA LOCALİZATİON**

Şimdi ise localization’ı controller da yapalım. UserController.cs adında bir controller oluşturalım ve ona giderek tanımlamasını yapalım.



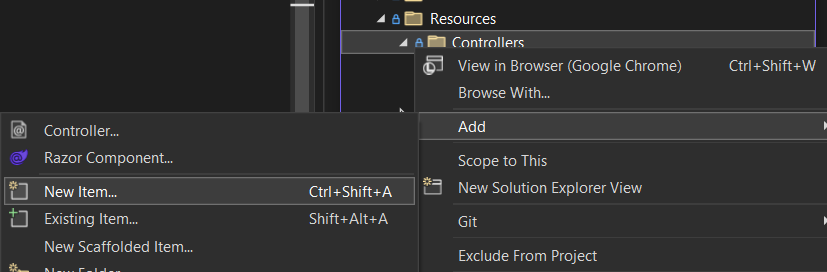
Localization kütüphanesini yukarı ekliyoruz.



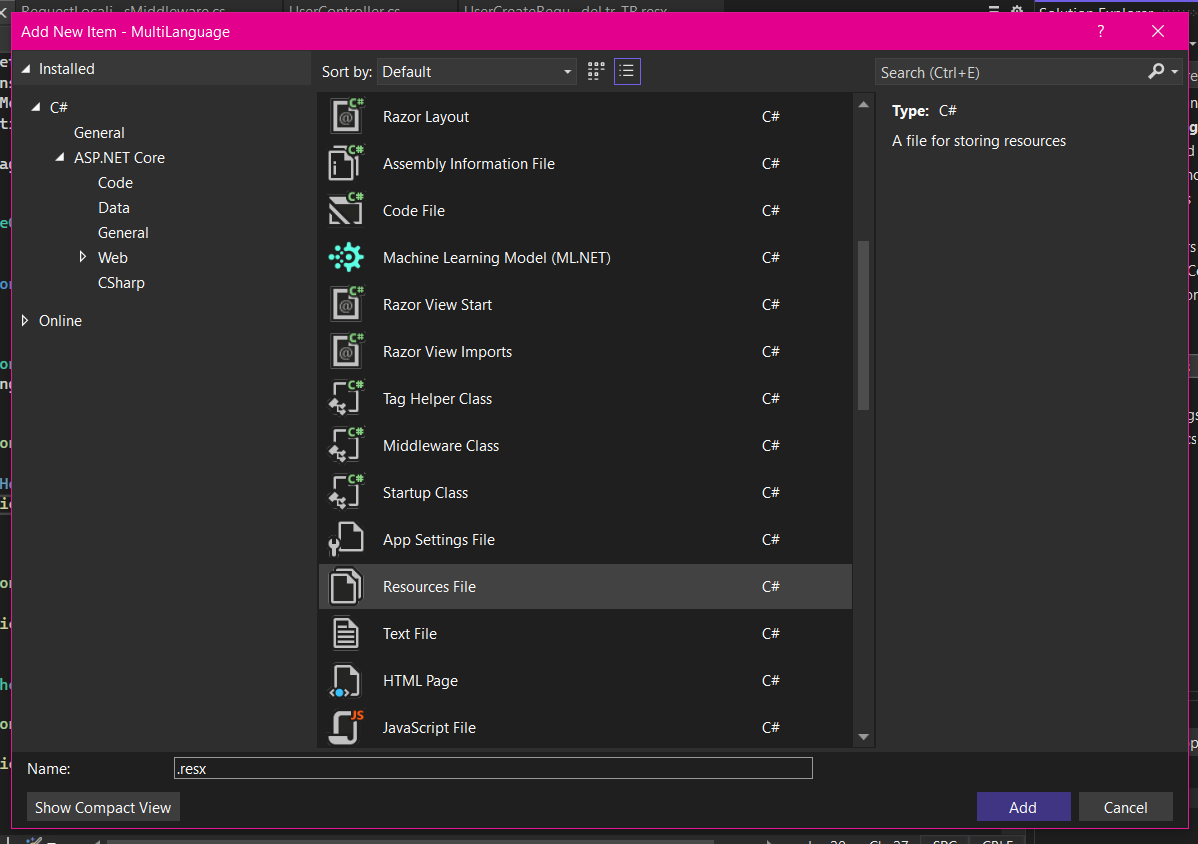
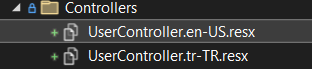
Sonrasında  dependency injection ile talep ediyoruz.



Sonrasında Resources dosyamızda Controllers adlı klasör oluşturuyoruz. Visual studionun oluşturacağımız dosyayı okuyabilmesi için isimlendirme çok önemlidir. Sonra (resx) item’i oluşturmamız gerekiyor.



Dosyanın isimlendirilmesi şu şekilde yapılır; Controllers klasörünün altında item oluşturduğumuz için geriye sadece hangi controllerda ve dilde olduğunu söylemek kalıyor. Aşağıda nasıl isimlendirme yapılacağını gösterdim.

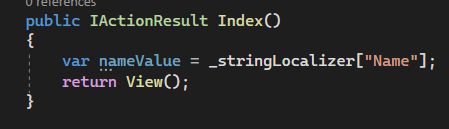
 

Sonrasında resx dosyalarımıza aşağıda ki gibi veri ekliyoruz.

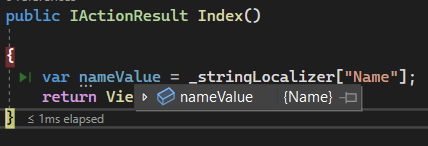




Sonra localizeri kullanmak istediğimizde UserController de aşağda ki şekilde tanımlamasını yapıyoruz. “Name” anahahtar kelimesini çağrıyoruz.

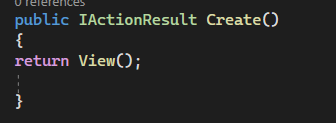
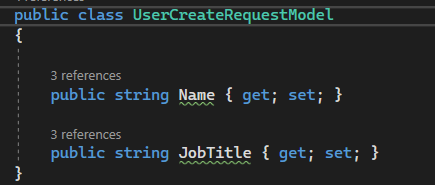


Projeyi çalıştırdığımızda aşağıda ki şekilde debug koyup çalıştırdığımızda değerimizin geldiğini görüyoruz. İlk başlangıçta otomatik olarak bilgisayarımızın dilinde gelicektir.

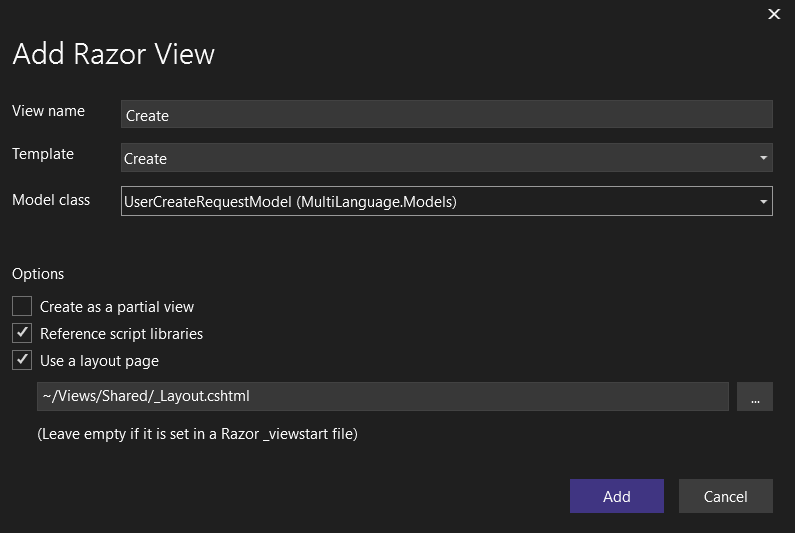


**MODELS’LERDE LOCALİZATİON**

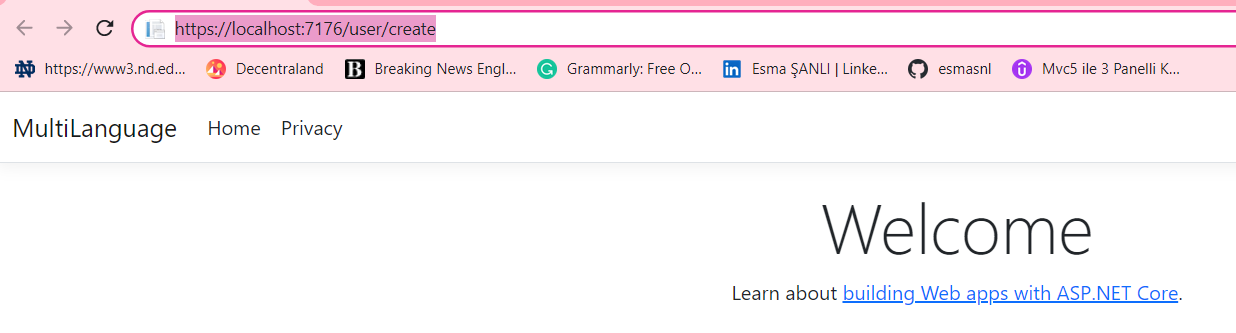
Resources dosyamızın altına Models adında klasör oluşturuyoruz. İlk olarak UserControllerımıza Create methodu ekliyoruz. Sonrasında Models klasörü altında UserCreateRequestModel.cs classı oluşturuyoruz. Ve içerisine iki tane değer ekliyoruz.

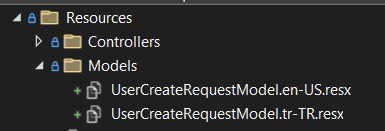
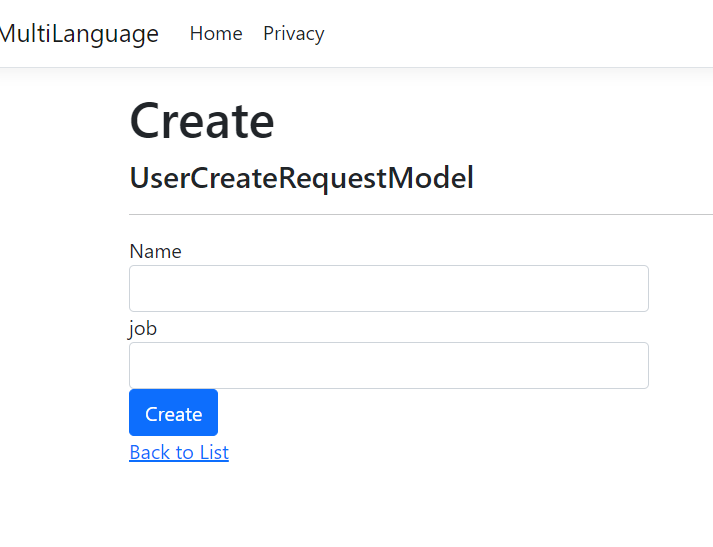
Sonrasında içerisinde UserCreateRequestModel sayfası olan Create sayfası oluşturmamız gerekiyor. Create methodumuzun üzerine gelip sağ tık yaparak add wiew seçeneğine tıklıyoruz. Ve aşağıda ki şekilde oluşturuyoruz.



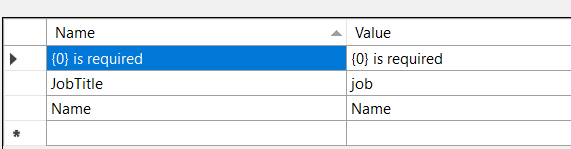
Sayfa oluştuktan sonar projemizi çalıştırdığımızda gelen sayfaya aşağıda ki şekilde url girerek enter’lıyoruz.

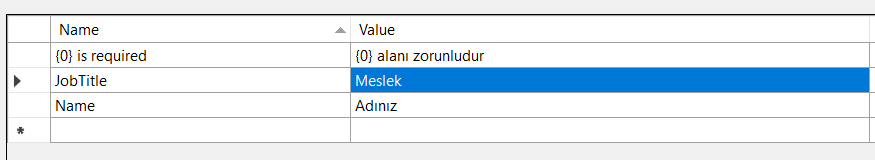


Karşımıza böyle bir sayfanın gelmesi gerekiyor. Şimdi ki amacımız ise name job kısmının altına kullanıcı boş bırakır ise boş geçilemez değerleri vermek ve bunları Türkçe, İngilizce şeklinde göstermek. Şimdi Resources dosyamızın altına models resx dosyalarımızı oluşturuyoruz.

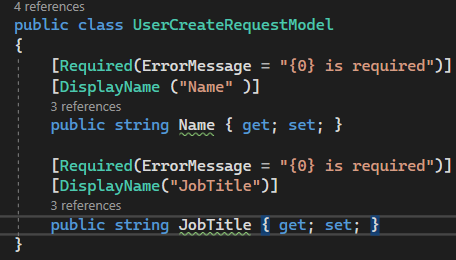
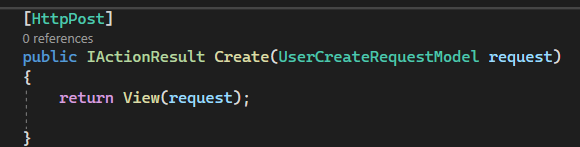


Resx dosyamıza dil seçeneğine göre aşağıda ki değerleri giriyoruz.

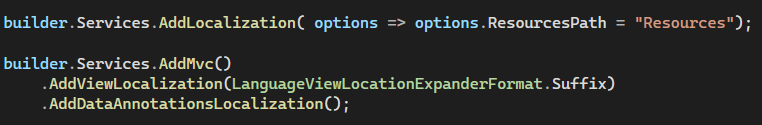




Ve sonrasında ise UserCreateRequestModel.cs dosyamıza gelerek aşağıda ki kodları ekliyoruz. Sonra UserControllera giderek aynı isimde HTTP method tanımlamamız gerekiyor. (CTRL . ile tanımlıyoruz.)

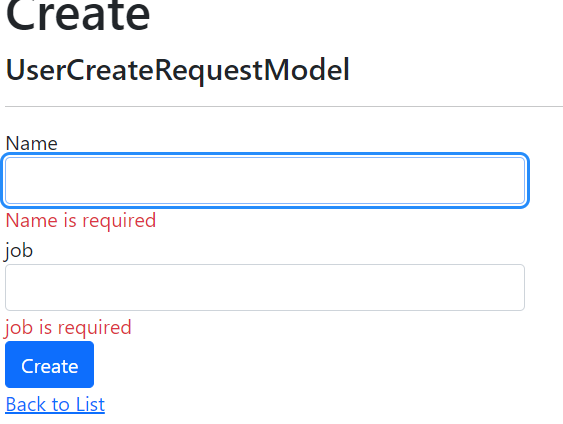
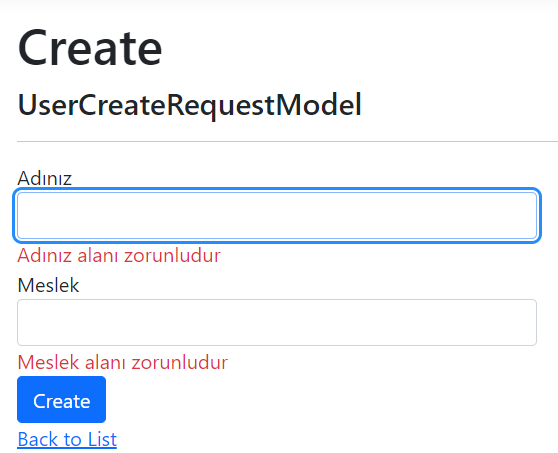
 

Şimdi ise program.cs dosyasında yapılandırmasını yapıyoruz.



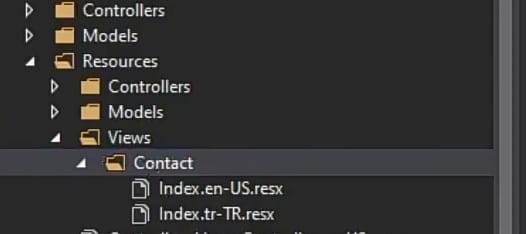
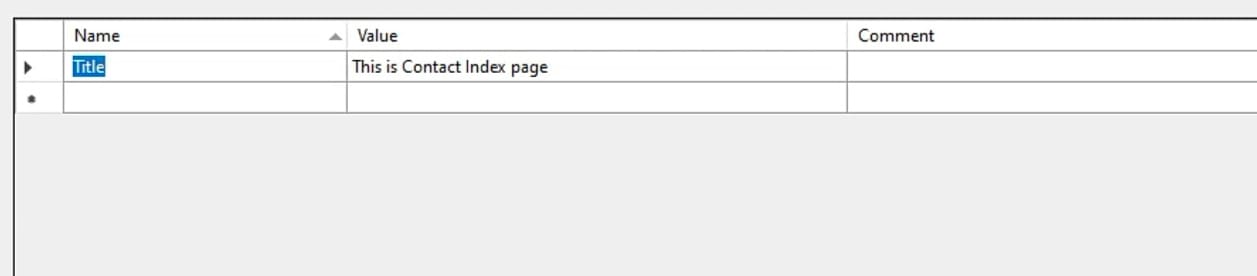
Sayfamızın aşağıda ki şekilde hataları vermesi gerekiyor. Ve url kısmına aşağıda ki şekilde dil değerini girdiğimizde değerlerin türkçe olması gerekiyor.

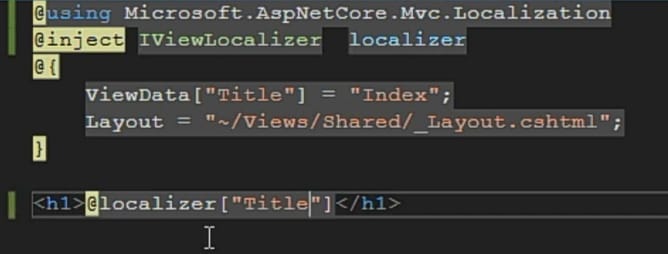


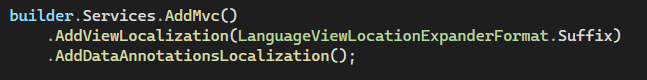
**VİEWS’LERDE LOCALİZATİON**

Yukarıda yapılan işlemlerin aynısını yeni bir controller oluşturup oradan view sayfası oluşturarak işlemleri aşağıda ki şekilde tekrarlayarak ve verdiğim kodlarını sayfanıza girerek, sonrasında ise program.cs içerisinde yapılandırmayı yaparak viewslerde ki localiationu oluşturabilirsiniz.

****  ****

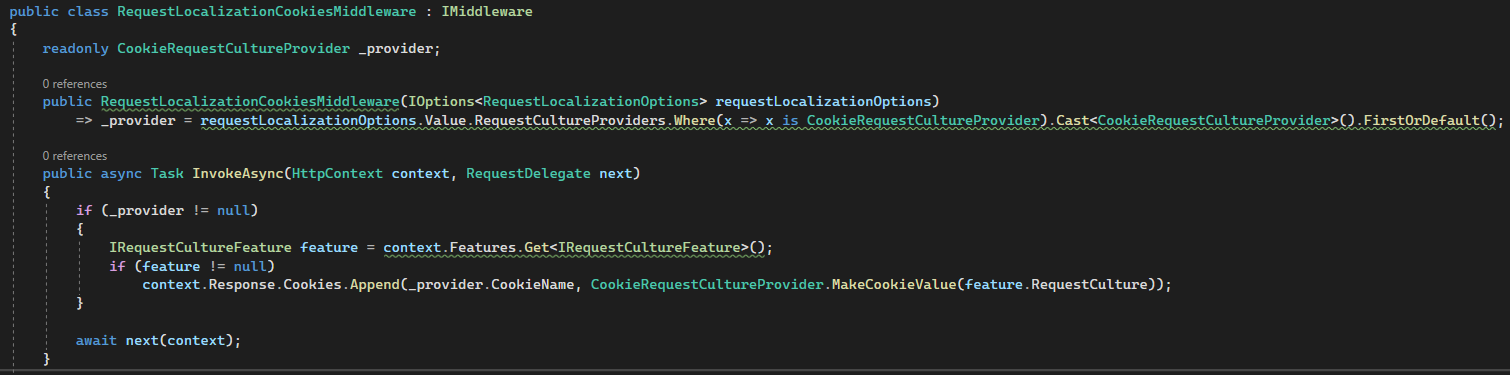
****

View sayfasının yapılandırmasını en başta yazdığımız kodlarda yapmıştık.( AddViewLocalization)

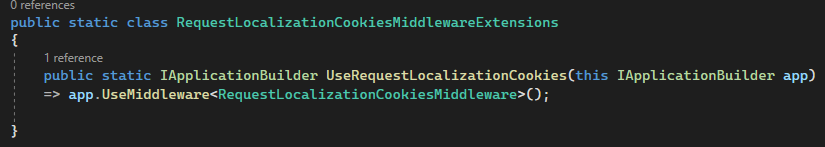


**CookieRequestCultureProvider İle Kullanıcı Kültürünü Yakalama**

İlk önce cookie değerini oluşturacak middlewar’i tanımlıyoruz. Bunu yeni bir sınıf oluşturarak yapıyoruz.



Sonrasın da bu middlewar’i entegre edecek olan extension methodunu oluşturuyoruz.



Oluşturduğumuz methodu program.cs de dependency injection container’ına ekleyecek kodu yazıyoruz.



Ve en son olarak ‘UseRequestLocalizationCookies’ extension metodunu çağıralım.



HAZIRLAYAN: ESMA ŞANLI