**Route (Yönlendirme):**

Bir web uygulamasında URL'lerin belirli bir kod veya işlemle eşleştirilmesini sağlayan bir yapıdır. Hem **frontend** (React gibi) hem de **backend** (ASP.NET Core gibi) tarafında kullanılabilir.

**Backend Tarafında Route İşlemi (ASP.NET Core):**

ASP.NET Core'da route işlemi, bir URL'yi belirli bir controller ve action metodu ile eşleştirmek için kullanılır.

**Route Kullanımının Faydaları**

* **Organizasyon**: URL'lerin belli bir düzenle çalışmasını sağlar.
* **Navigasyon**: Kullanıcılar arasında kolay gezinme imkânı sunar.
* **API Entegrasyonu**: Backend'de farklı endpoint'ler oluşturabilir ve frontend'de onlara erişim sağlayabilirsiniz.

**Route Çeşitleri**

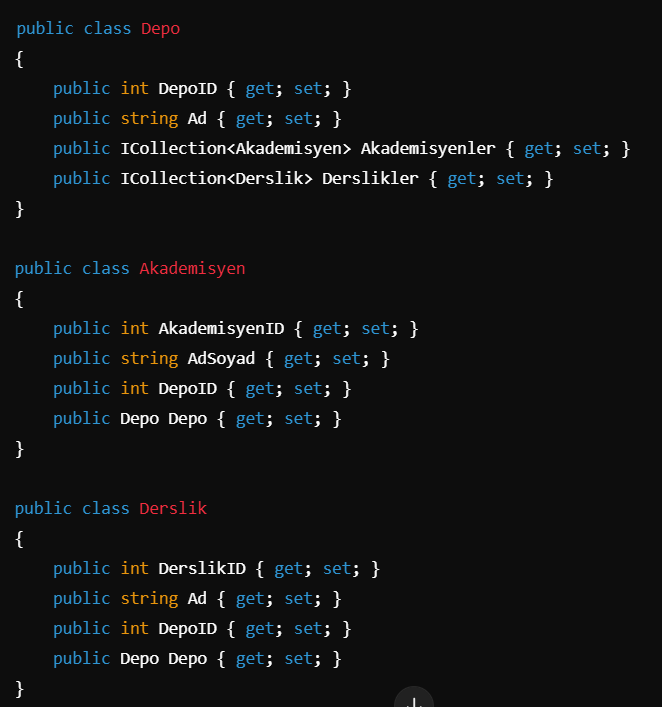
1. **Conventional Routing (Geleneksel Yönlendirme)**
2. **Attribute Routing (Özellik Yönlendirme)**
3. **Endpoint Routing**
4. **Custom Routing (Özel Yönlendirme)**
5. **Area Routing**
6. **API Routing (RESTful Yönlendirme)**
7. **Dynamic Routing (Dinamik Yönlendirme)**
8. **Blazor Routing**

Resource isimlendirmesi çoğul(**customers**) veya tekil(**customer**) şeklinde olabilir. Çoğul olarak erişilmek istenirse /c**ustomers**veya detay bazlı erişilmek istenirse(yani tekil olarak)**/customers/{id}** ile erişebilceğinden çoğul kullanılması önerilir.İsmimlendirmeye Get, Insert, Update gibi actionlar dahil edilmemelidir.

h[ttp://www.example.com/customers](http://www.example.com/customers)/{customerId}/orders/{orderId}

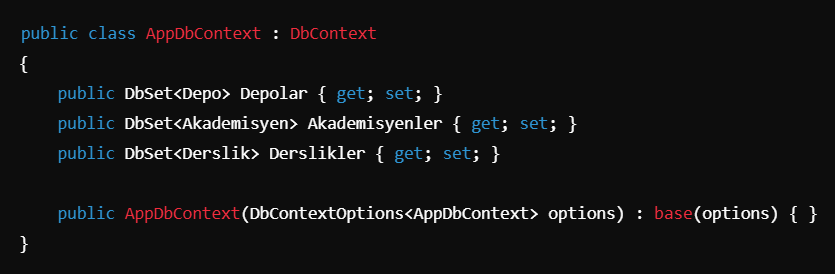
**Projemizde Akademisyenler için Route Yapısı Oluşturma**

**1.** **Veritabanı Tasarımı:**

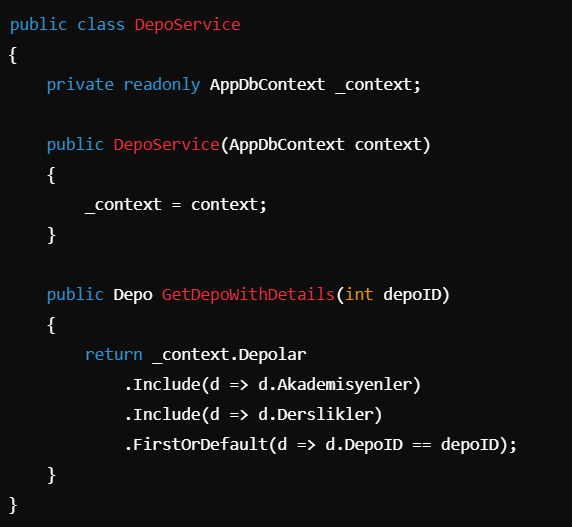
****

**2.** **ASP.NET Core API Yapısını Kurma:**

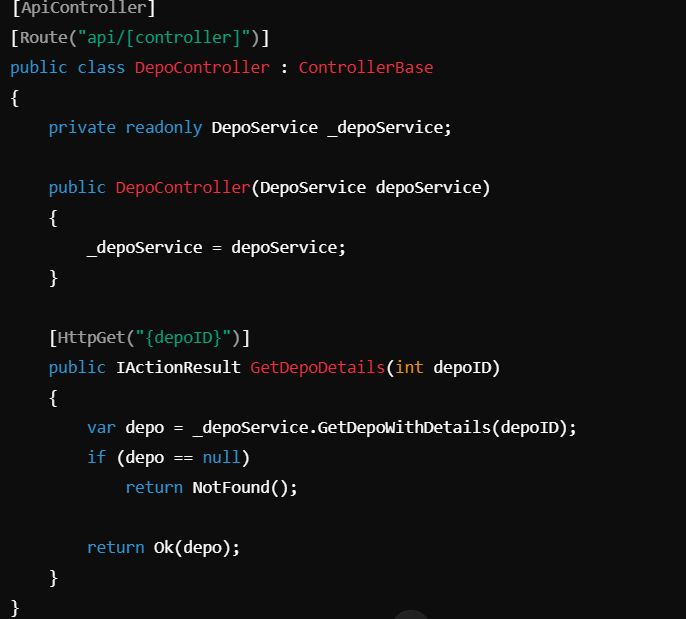
Veri tabanına bağlandıktan sonra DbContext oluştur:

****

**Veritabanı Servislerini Yazma(**Akademisyenler ve derslikler için gerekli veriyi almak için servis sınıflarını oluştur):

****

**Controller Oluşturma** (API controller'ını yazalım):

****

(Bu, belirli bir depoya ait akademisyenler ve dersliklerin verisini döndüren bir API sağlar)

**3.** **React Uygulamasında Route Yapısını Kurma**

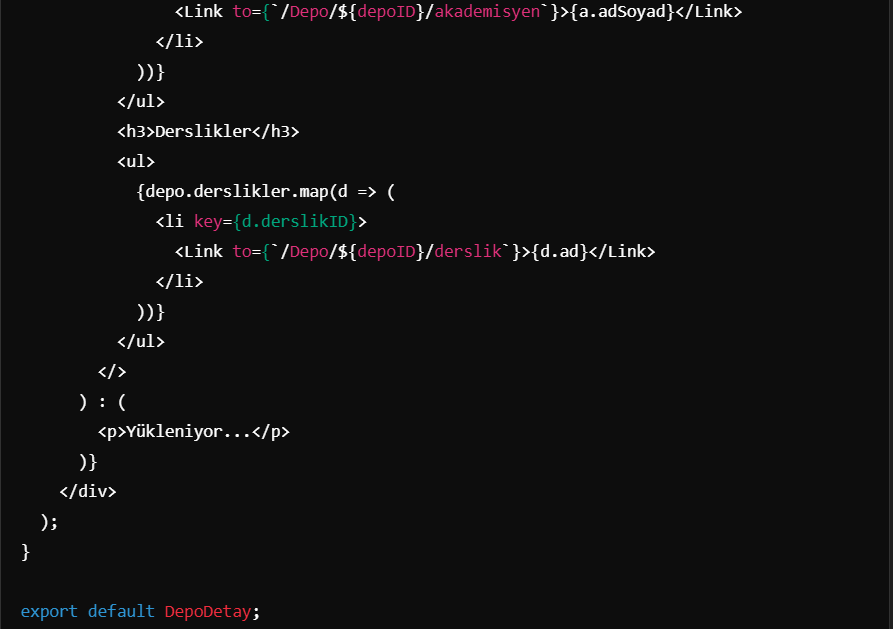
**React Router Yapısını Kurma** (React Router kullanarak yönlendirmeleri tanımlayalım. React Router'ı kurduktan sonra aşağıdaki gibi bir yapı oluşturabilirsiniz):

****

**Route Yapısını Tanımlama**

****

DepoDetay sayfasında, depo ID’sini alarak API'den verileri çekebilirsiniz:

**** ****

### ****SONUÇ-ÖZET****

* **Backend (ASP.NET Core)**:
  + Veritabanı yapısını Entity Framework Core ile kurdunuz.
  + RESTful API'ler oluşturularak React frontend'e veri sağladınız.
  + API controller'ları ile dinamik veriler alınıp sunuldu.
* **Frontend (React)**:
  + React Router kullanarak, URL yapıları dinamik hale getirildi.
  + depoID gibi parametreler kullanılarak, her depo için ayrı sayfalar oluşturuldu.
  + Depo detay sayfasında, akademisyenler ve derslikler gibi alt sayfalara yönlendirme yapıldı

**Kaynakça:**[**https://feyyazacet.medium.com/asp-net-core-web-api-ile-ger%C3%A7ek-anlamda-restful-servisler-geli%C5%9Ftirmek-b9d496789ab7**](https://feyyazacet.medium.com/asp-net-core-web-api-ile-ger%C3%A7ek-anlamda-restful-servisler-geli%C5%9Ftirmek-b9d496789ab7)

[**https://learn.microsoft.com/en-us/aspnet/web-api/overview/web-api-routing-and-actions/routing-in-aspnet-web-api**](https://learn.microsoft.com/en-us/aspnet/web-api/overview/web-api-routing-and-actions/routing-in-aspnet-web-api)

[**https://learn.microsoft.com/en-us/aspnet/core/fundamentals/routing?view=aspnetcore-9.0**](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/routing?view=aspnetcore-9.0)

[**https://chatgpt.com/share/67754f64-7028-800e-830e-1d0de6cbf6a4**](https://chatgpt.com/share/67754f64-7028-800e-830e-1d0de6cbf6a4)