



# Asp.Net ile Web Programlama

Doç. Dr. Zafer CÖMERT



**Bölüm  
4**

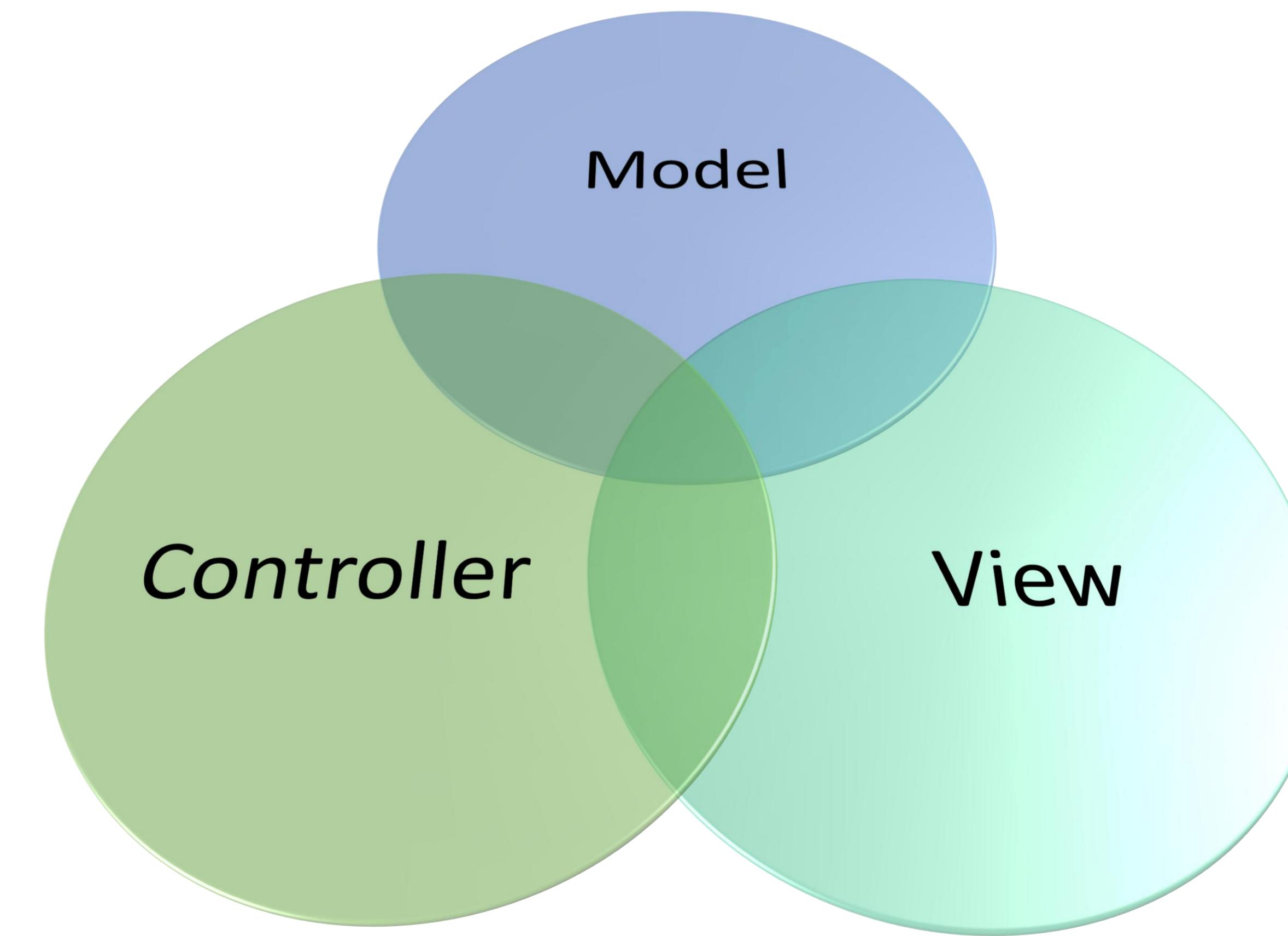
**MVC Temelleri**

# Giriş

## İçerik

- MVC Mimari Deseni
- Models
- Views
- Controllers
- Tag Helpers
- Dependency Injection
- Model Binding
- Model Validation

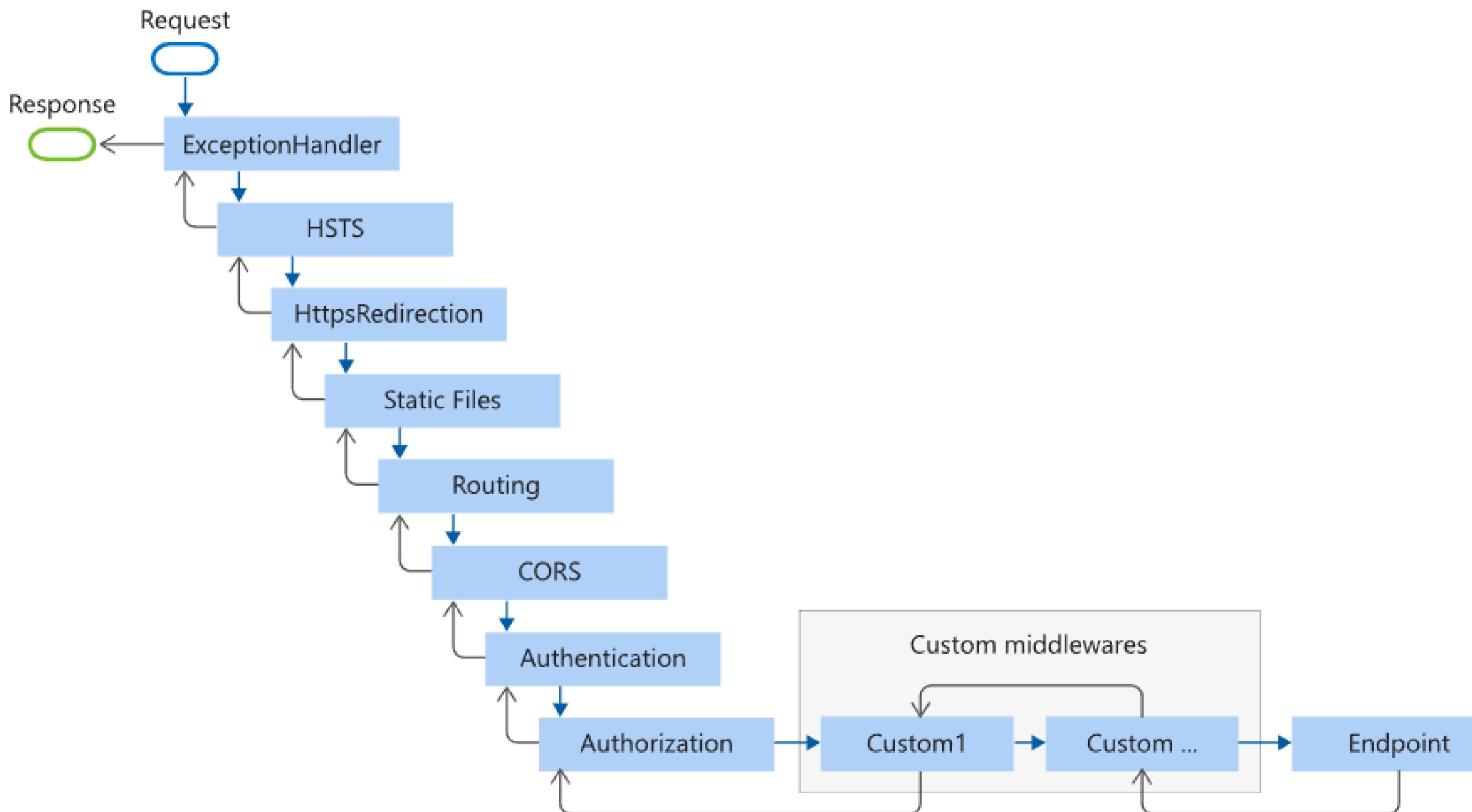
MVC

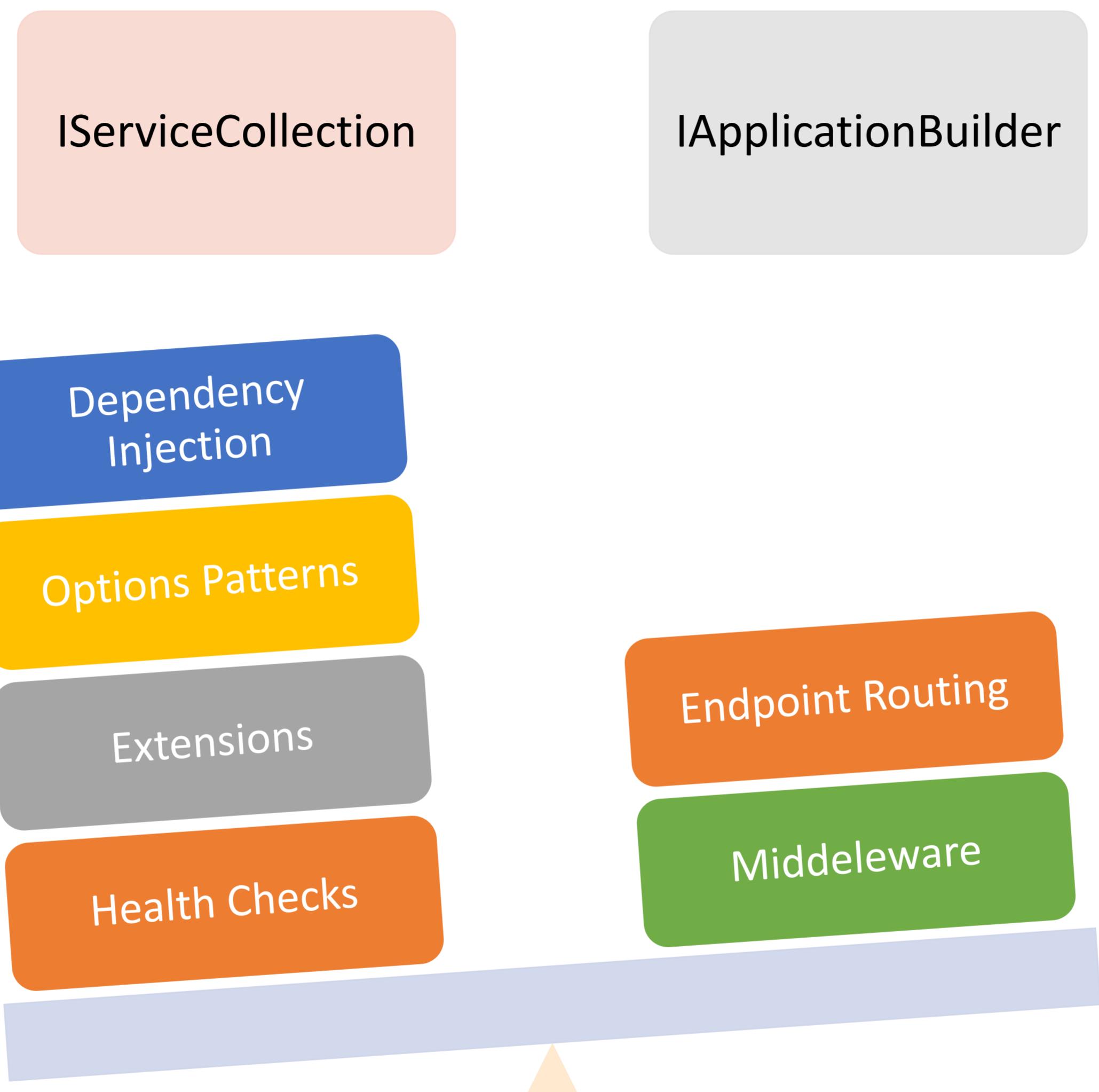


MVC  
Template

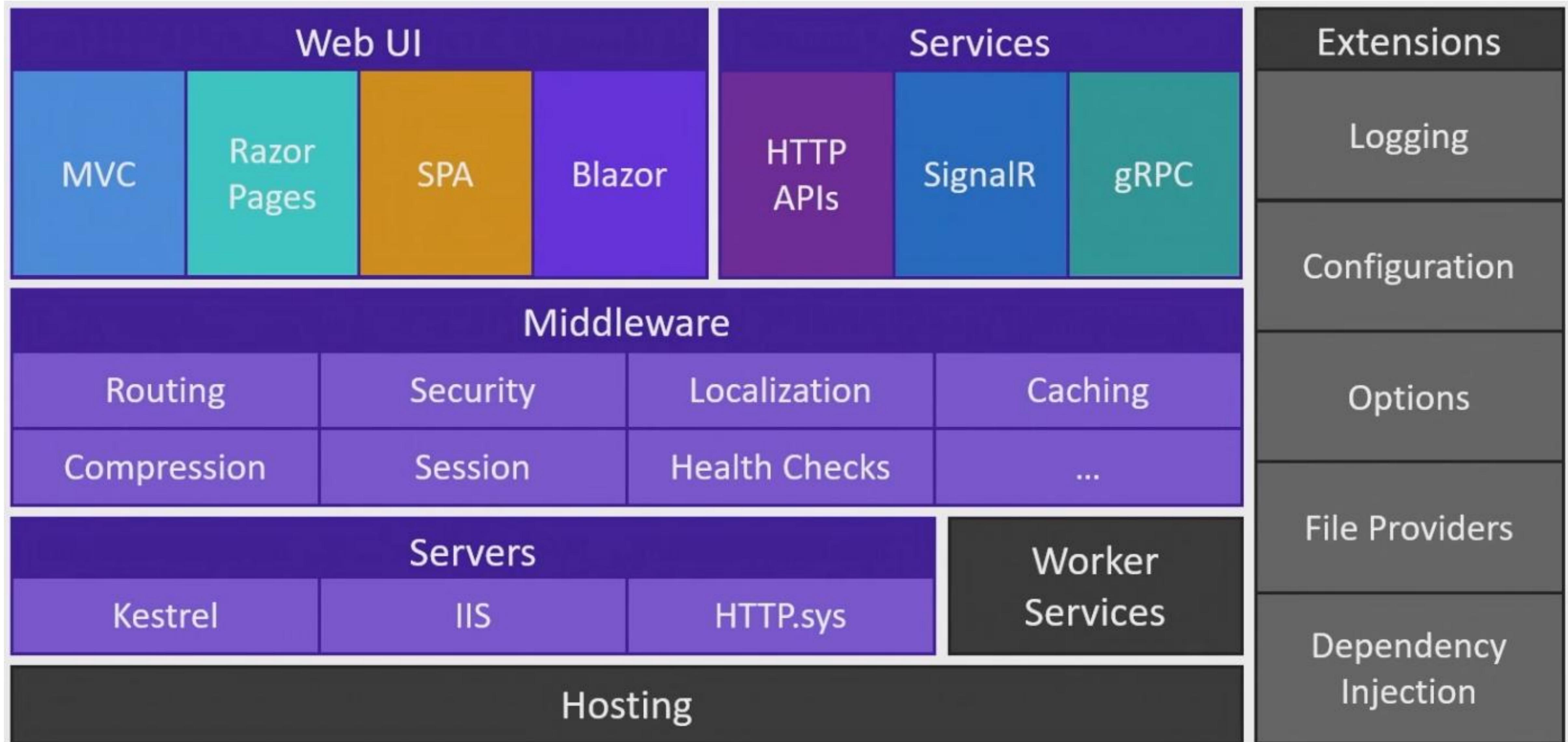
```
> dotnet new mvc -o ContactApp
```

# Pipeline





IServiceCollection'da Servisler Dependency Injection ile yönetilir. IApplicationBuilder ise Middleware zincirleri kurmak için kullanılır.



\*

# Katmanlı Mimari ve MVC

Özellik	Katmanlı Mimari (N-Tier)	MVC (Model–View–Controller)
<b>Amacı</b>	Uygulamanın tüm katmanlarını (sunum, iş mantığı, veri erişim) birbirinden ayırmak	Kullanıcı arayüzü (UI) içinde veri, görünüm ve kontrolü ayırmak
<b>Odak Noktası</b>	Fiziksel ve mantıksal ayrımlar (sunucu, servis, veritabanı)	UI düzeni ve kullanıcı etkileşimi
<b>Katmanlar/Bileşenler</b>	- Presentation (Sunum)- Business Logic (İş Mantığı)- Data Access (Veri Erişim)	- Model- View- Controller
<b>Kapsam</b>	Tüm uygulama mimarisi (geniş kapsamlı)	Sunum katmanı içi düzenleme (daha dar kapsamlı)
<b>Uygulama Yeri</b>	Büyük projelerde farklı DLL/proje olarak ayrılabilir	Tek bir web uygulaması içinde (ASP.NET Core MVC gibi)
<b>Avantajları</b>	- Yeniden kullanılabılırlik- Bakım kolaylığı- Katmanlar arası bağımsızlık	- Ayrık sorumluluk (Separation of Concerns)- Test kolaylığı- UI geliştirmede hız ve düzen
<b>Örnek Senaryo</b>	Öğrenci bilgi sistemi: Veritabanı erişimi için ayrı Data katmanı, iş kuralları için Business katmanı, UI için Presentation katmanı	Sunum katmanı içinde: Controller isteği alır, Model'den veriyi çeker, View ile kullanıcıya gösterir

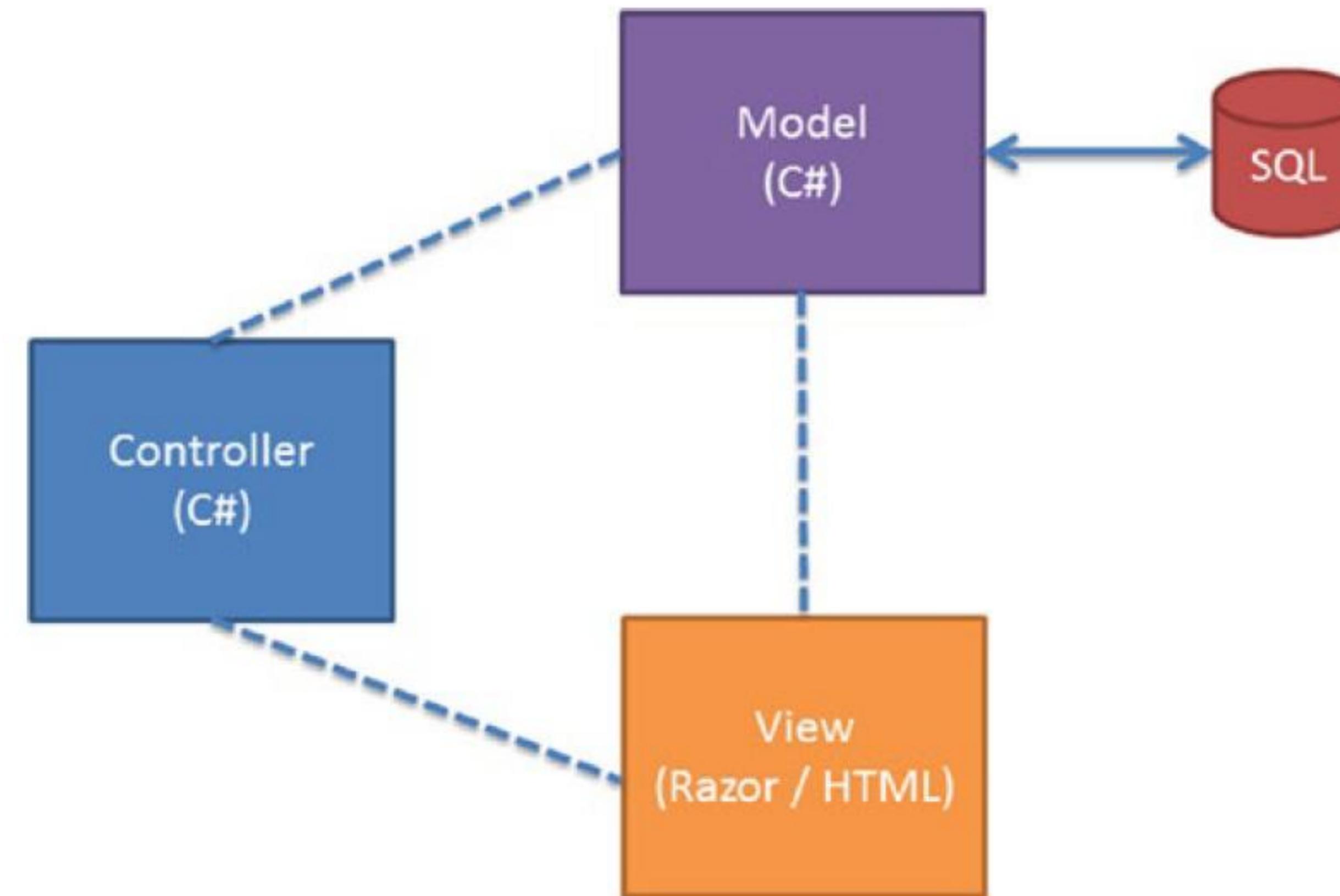
# Coding by Convention

# Coding by Convention

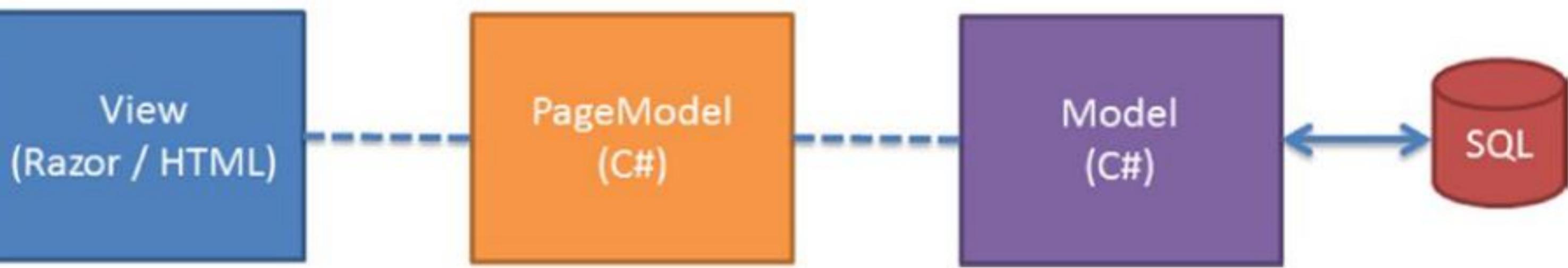
```
GuitarShop
  /Controllers
    /HomeController.cs
    /ProductController.cs
  /Models
    /Product.cs
  /Views
    /Home
      /About.cshtml
      /Index.cshtml
    /Product
      /Detail.cshtml
      /List.cshtml
  /wwwroot
    /css
      custom.css
    /images
    /js
      custom.js
    /lib
      /bootstrap
      /jquery
  /Startup.cs
```

## MVC

- MVC, mimarisel bir tasarım desenidir.

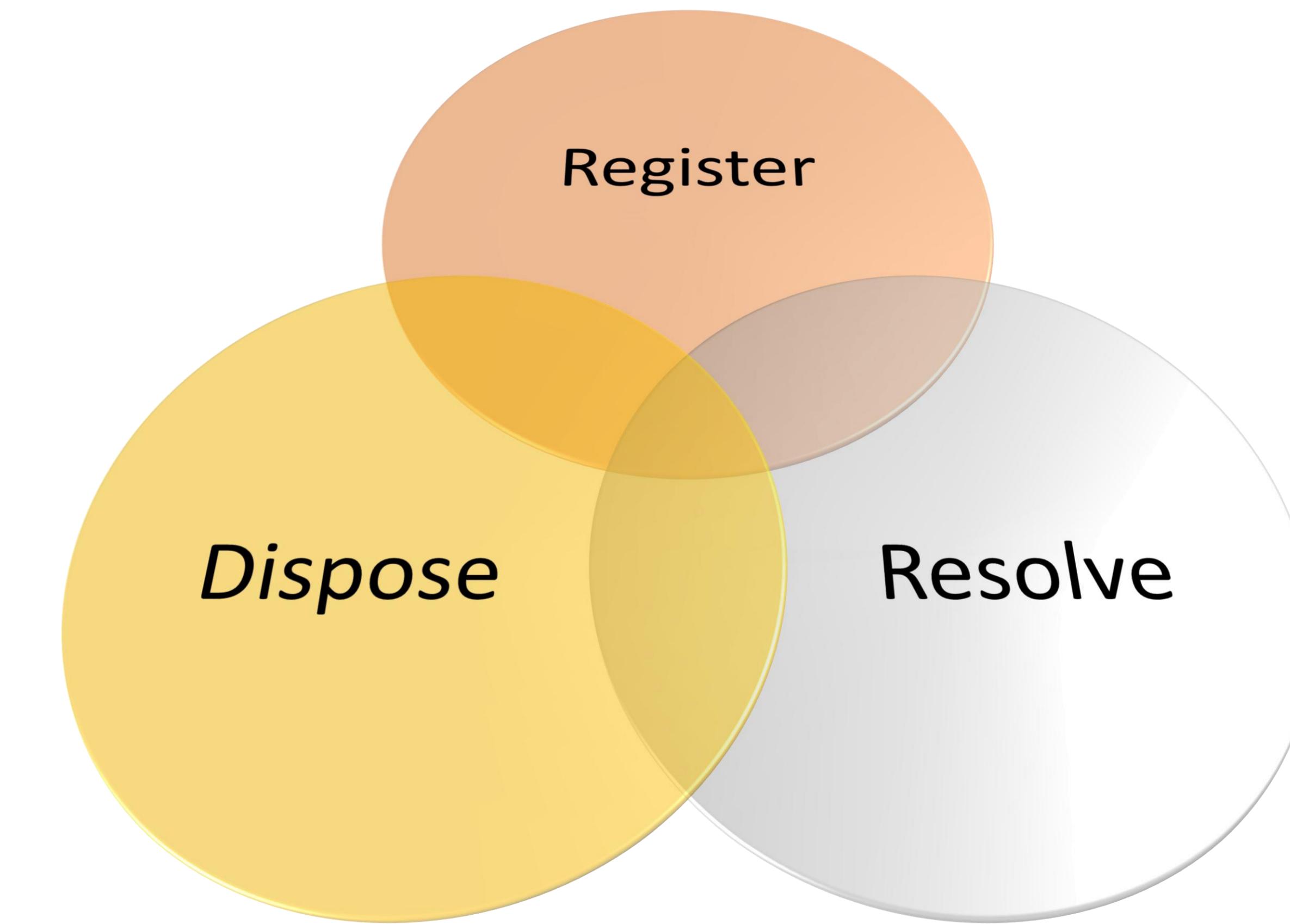


# MVVM



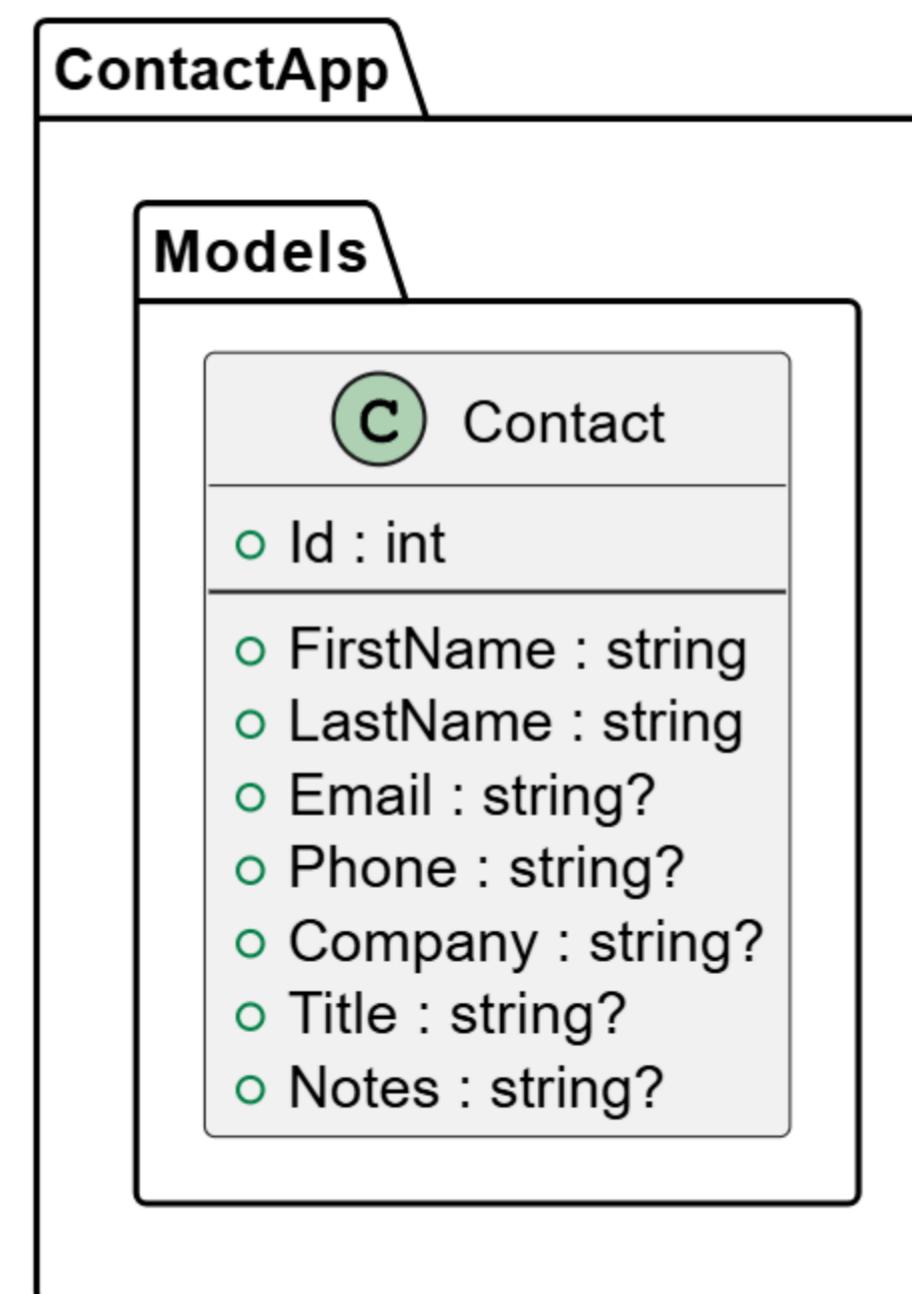
- Model-View-View-Model deseni

## Dependency Injection



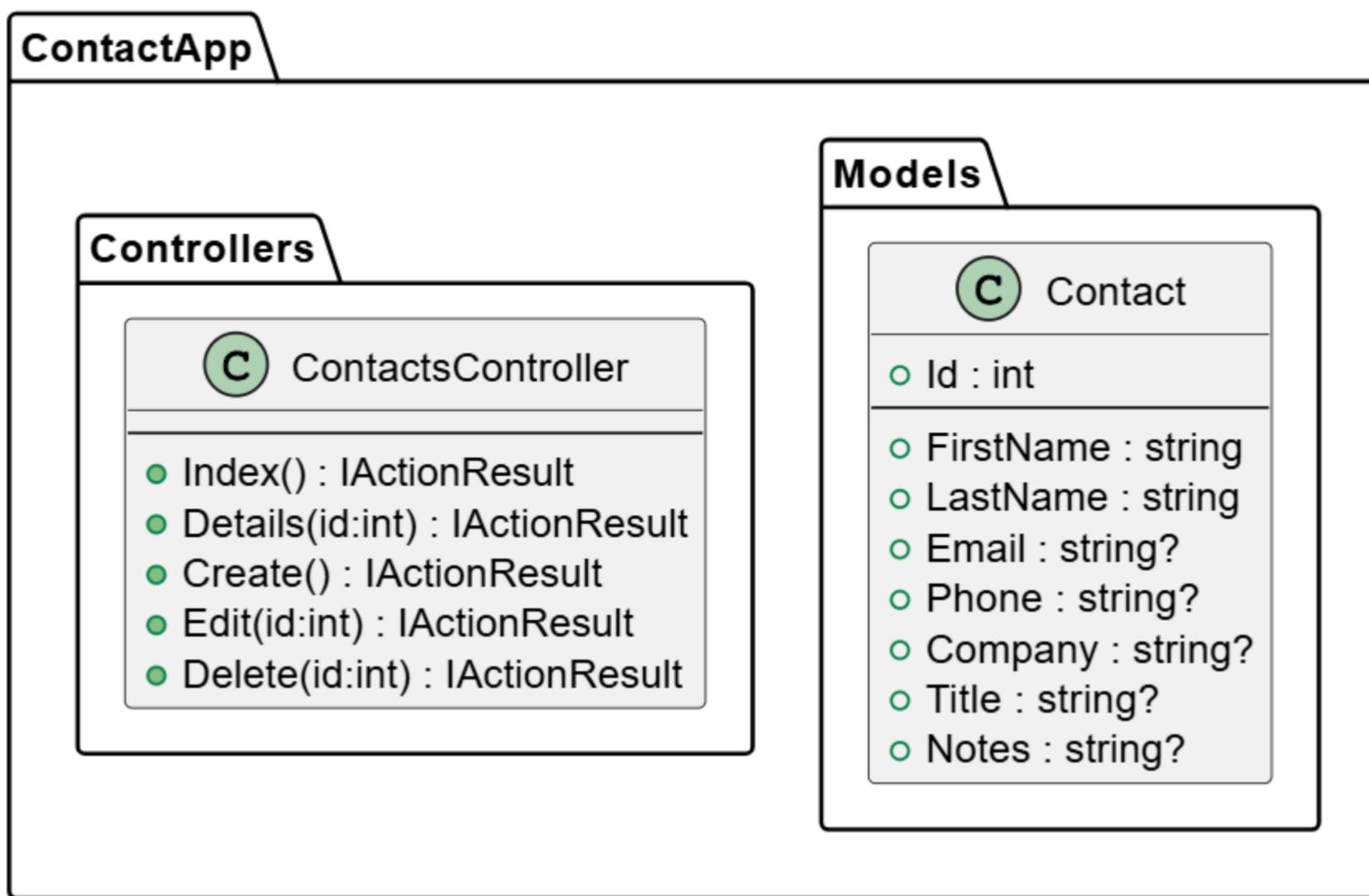
# Models

ContactApp.Models - Contact Sınıf Diyagramı

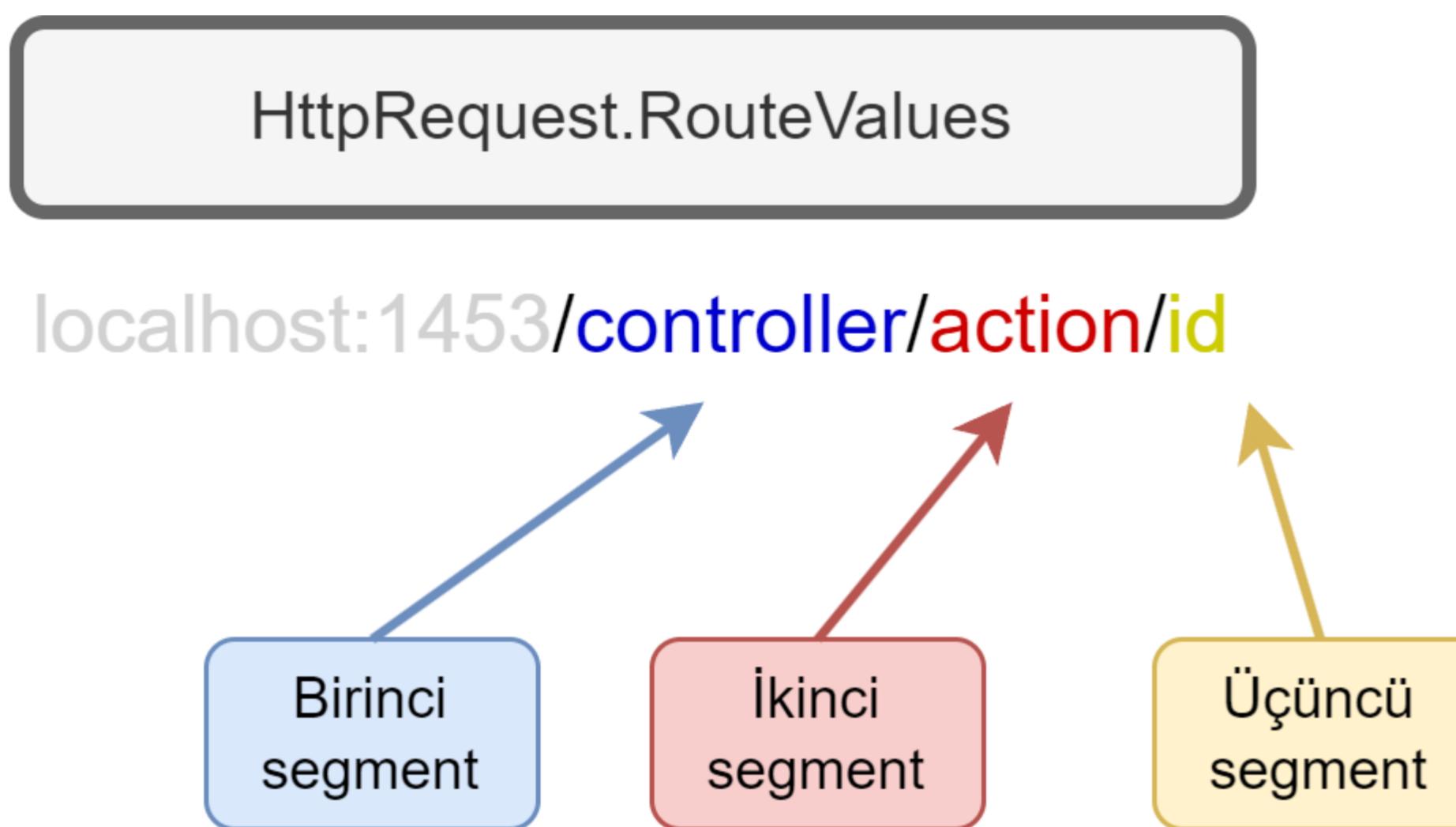


# Models - Controllers

ContactApp.Models - Contact Sınıf Diyagramı



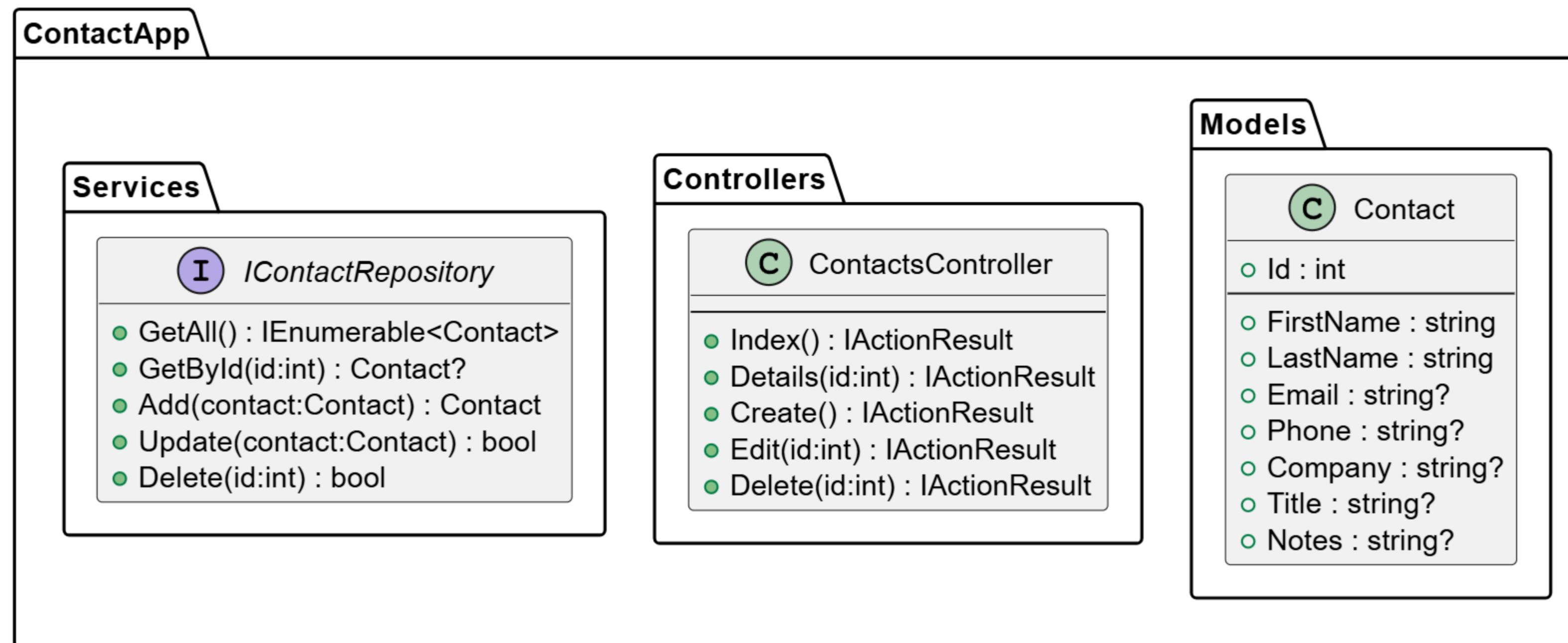
# Routing



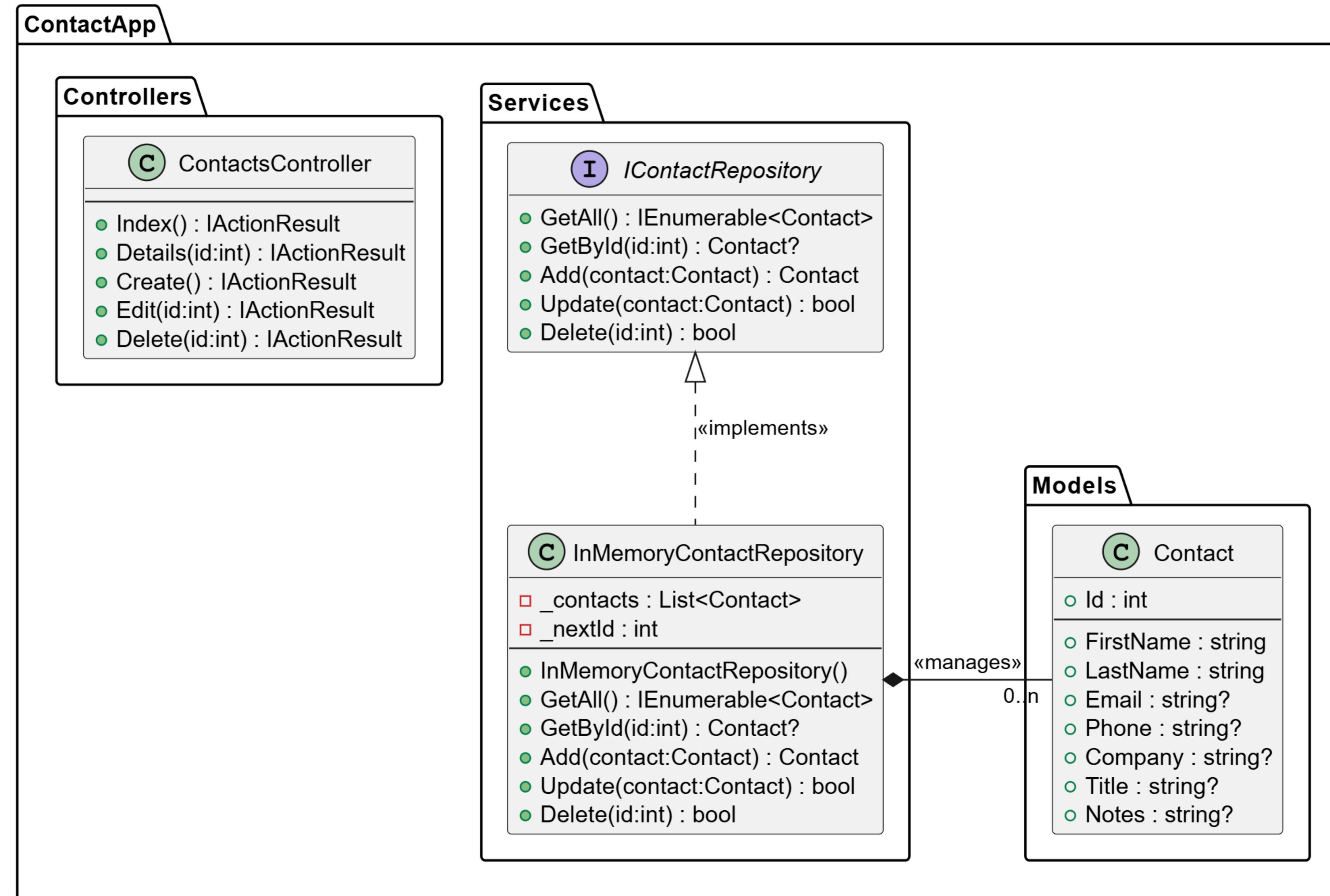
```
1 app.MapControllerRoute(  
2     name: "default",  
3     pattern: "{controller=Home}/{action=Index}/{id?}")  
4     .WithStaticAssets();
```

# Models - Controllers

ContactApp.Models - Contact Sınıf Diyagramı



## ContactApp.Models - Contact Sınıf Diyagramı



# Tag Helpers

TagHelperContext

TagHelperOutput

TagHelperContent

- Tag Helpers, ASP.NET Core MVC ve Razor Pages ekosisteminde HTML etiketleri ile .NET tabanlı sunucu tarafı kodun bütünlüğünü sağlayan güçlü bir özelliktir.

- Tag Helpers doğrudan HTML etiketleri üzerinde çalışır.

# Tag Helpers

```
<form asp-action="...">
```

```
<input asp-for="PropertyName" />
```

```
<label asp-for="PropertyName"></label>
```

```
<span asp-validation-for="PropertyName"></span>
```

```
<div asp-validation-summary="All"></div>
```

```
<a asp-controller="..." asp-action="..." asp-route-id="...">
```

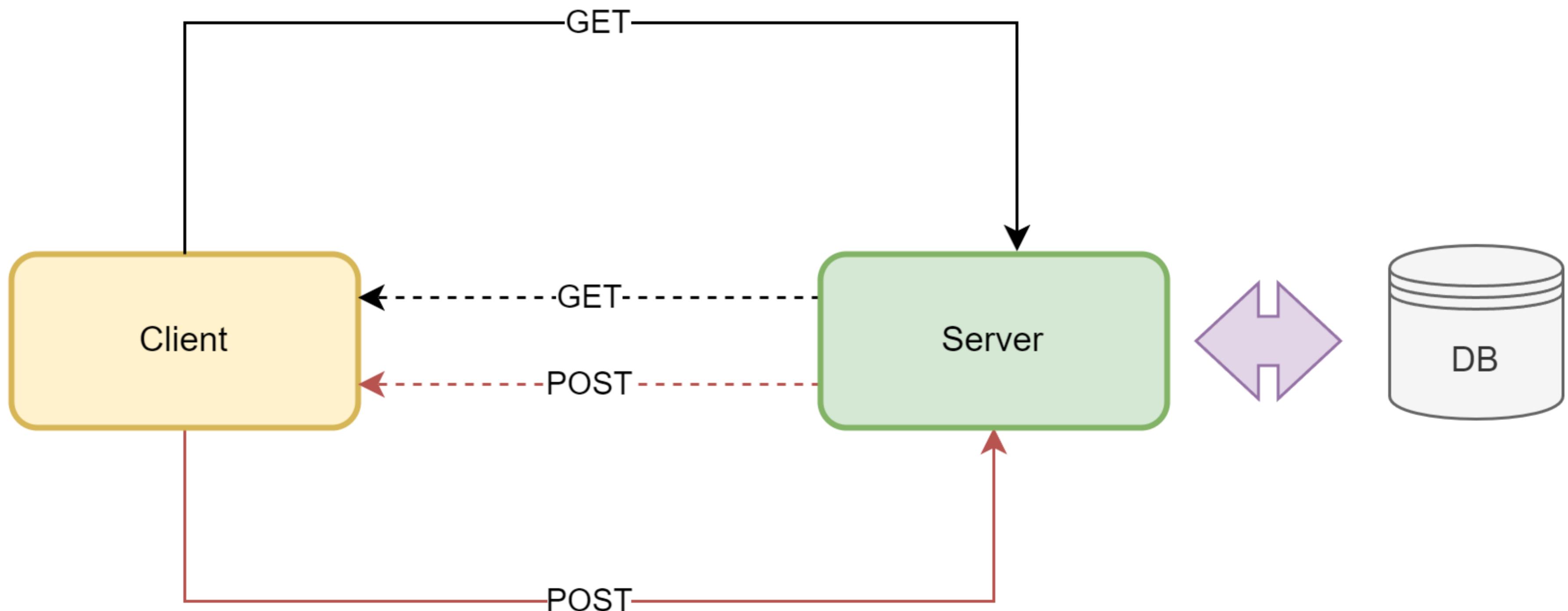
```
<select asp-for="PropertyName" asp-items="Model.ItemList"></select>
```

Razor  
Template

```
> dotnet new webapi -f net9
```

# PRG Pattern

## Post-Redirect-Get (PRG) Pattern



# Coding Over Configuration

Brand

Search

Left Bar Menu

Render Body()

Logo

Copyright

Social Media

# Teşekkürler

ZAFER CÖMERT  
Öğretim Üyesi