

Controller ve View Arasındaki İletişim

Controllerdan view'a veri nasıl göndeririz?

ViewData/ ViewBag/ TempData

Dr. Öğr. Üyesi Fatma AKALIN

Controller'dan View'a Veri Gönderme

Controller'a eklediğimiz Action metodu üzerinden View oluşturma işleminin nasıl yapıldığını öğrenmiştik. Bazı durumlarda controller'dan view'a View oluştururken veri gönderilmesi gerekebilir.

Örneğin online satış sitesinde ürün sayfasında(ürün bilgileri, ürün resimleri ..) verileri controller tarafından alıp View a göndermemiz gerekebilir. Bu durumda farklı yöntemler ile bu verileri view'a nasıl aktaracağımızı öğreneceğiz. Bu işlemi başta bahsettiğimiz action metodu vasıtasıyla gerçekleştireceğiz.

Controllerdan viewa veri gönderme işlemi viewbag ve viewdata aracılığı ile yapılmaktadır. Bu bölümde viewbag ve viewdata kavramlarının neler olduğunu örnek uygulamalarla öğreneceğiz.

Dr. Öğr.

ViewData

ViewData, sözlük yapısı gibi anahtar değer ikilileri tutarak (aynı anahtardan iki tane olamıyor, olursa bir önceki ezilmiş olur) istediğimiz dataların viewa taşınmasında görev alan bir özellik. Peki biz viewdata'ya nasıl bir yükleme yapıyoruz?

Dr. Öğr. Üyesi Fethiye AKKALIN

Yeni projenizi yapılandırın

ASP.NET Web Uygulaması (.NET Framework)

Proje adı

denemesekizinciders

Konum

D:\aspnet\1_ADESLER

Çözüm adı ⓘ

denemesekizinciders

☐ Çözümü ve projeyi aynı dizine yerleştirin

Altyapı

.NET Framework 4.6.2

Proje "D:\aspnet\1_ADESLER\denemesekizinciders\denemesekizinciders"

Yeni ASP.NET Web Uygulaması oluşturun



Boş

ASP.NET uygulamaları oluşturmak için boş bir proje şablonu. Bu şablonda içerik yoktur.



Web Forms

ASP.NET Web Forms uygulamaları oluşturmak için bir proje şablonu. ASP.NET Web Forms, bilinen sürük ve bırak yöntemiyle olay denetimli modeli kullanarak dinamik web siteleri oluşturmanızı sağlar. Tasarım yüzeyi ile yüzlerce denetim ve bileşen, veri erişimi olan gelişmiş, güçlü, kullanıcı arabirimi denetimli web sitelerini hızlı bir şekilde oluşturmanızı sağlar.



MVC

ASP.NET MVC uygulamaları oluşturmaya yarayan bir proje şablonu. ASP.NET MVC, Model-View-Controller mimarisini kullanarak uygulamalar oluşturmanıza olanak sağlar. ASP.NET MVC en son standartları kullanan uygulamalar oluşturmak için hızlı, test güdümlü geliştirmeye olanak sağlayan birçok özellik içerir.



Web API

Tarayıcılar ve mobil cihazlar dahil olmak üzere çok çeşitli istemcilere erişebilen RESTful HTTP hizmetleri oluşturmaya yarayan bir proje şablonudur.



Tek Sayfalı Uygulama

ASP.NET Web API kullanarak JavaScript temelli zengin istemci tarafı HTML5 uygulamaları oluşturmak üzere kullanılan proje şablonudur. Tek Sayfalı Uygulamalar, HTML5, CSS3 ve JavaScript'in kullanıldığı istemci tarafı etkileşimler içeren zengin bir kullanıcı deneyimi sağlar.

Kimlik Doğrulama

Yok

Klasörleri ve çekirdek başvurularını ekle

☐ Web Forms

☒ MVC

☐ Web API'si

Gelişmiş

☒ HTTPS'yi Yapılandır

☐ Docker desteği

(Docker Desktop gerektiriyor)

☐ Ayrıca birim testleri için bir proje oluşturun

denemesekizinciders.Tests

Geri

Oluştur

Yeni İskeleli Öğe Ekle

Yüklü

Ortak

MVC

Alan

Denetleyici

Göster

Web API'si

Entity Framework kullanan, görünümleri olan MVC 5 Denetleyici

MVC 5 Denetleyici - Boş

Okuma/yazma eylemleri olan MVC 5 Denetleyici

MVC 5 Denetleyici - Boş
şunun tarafından Microsoft
v5.0.0.0
Boş bir MVC denetleyici.
Kimlik: MvcControllerEmptyScaffolder

Denetleyici Ekle

Denetleyici adı:

HomeController

Ekle

İptal

```
public class HomeController : Controller
{
```

```
// GET: Home
```

```
0 başvuru
```

```
public ActionResult Index()
```

```
{
```

```
    ViewData["text1"] = "Fatma Akalın";
```

```
    ViewData["check1"] = true;
```

```
    return View();
```

```
}
```

```
namespace denemesekizinciders.Controllers
```

```
{
```

```
0 başvuru
```

```
public class HomeController : Controller
```

```
{
```

```
// GET: Home
```

```
0 başvuru
```

```
public ActionResult Index()
```

```
{
```

```
    ViewData["text1"] = "Fatma Akalın";
```

```
    ViewData["check1"] = true;
```

```
    return View();
```

```
}
```

Yükü

Ortak

MVC

Alan

Denetleyici

Göster

Web API'si



MVC 5 Görünümü

MVC 5 Görünümü
şunun tarafın
v5.0.0.0

Türü kesin be
Model olma

Görünüm Ekle

Görünüm adı:

Index

Şablon:

Empty (model olmadan)

Model sınıfı:

Seçenekler:

☐ Kısmi görünüm olarak oluştur

☒ Başvuru betik kitaplıkları

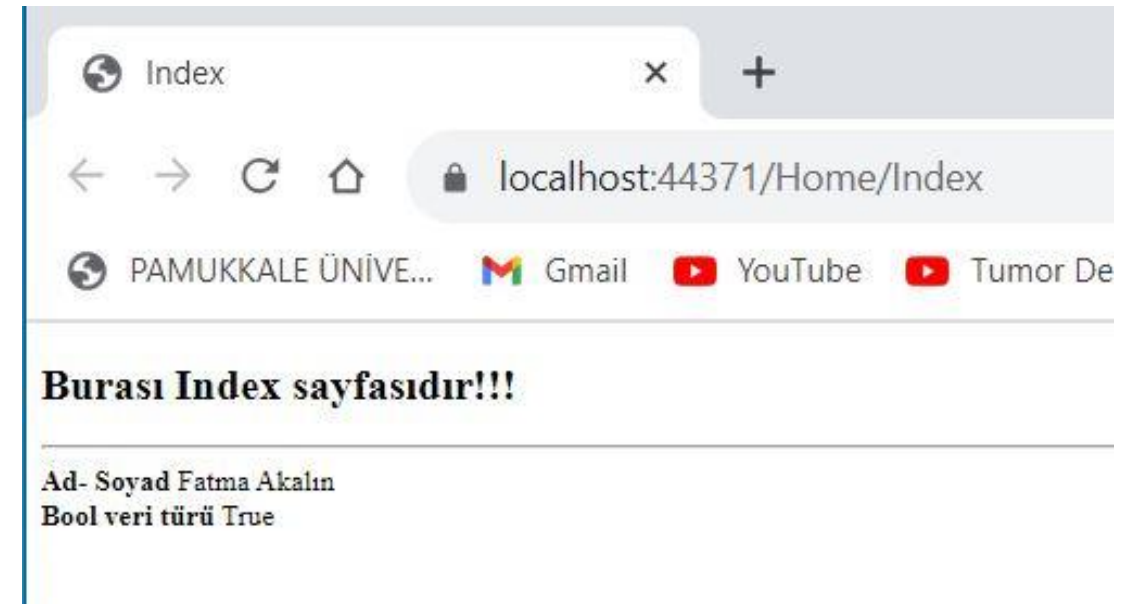
☐ Bir düzen sayfası kullanın:

(Bir Razor _viewstart dosyasında ayarlandıysa, boş bırakın)

Ekle

İptal

```
Index.cshtml  HomeController.cs
8  <html>
9  <head>
10     <meta name="viewport" content="width=device-width" />
11     <title>Index</title>
12 </head>
13 <body>
14     <div>
15         <h2>Burası Index sayfasıdır!!!</h2>
16         <hr />
17     </div>
18     <div>
19         <b>Ad- Soyad</b> @ViewData["text1"]
20         <br />
21         <b>Bool veri türü</b> @ViewData["check1"].ToString()
22     </div>
23 </body>
24 </html>
25
```



ViewBag

Proje kapsamında oluşturulan aynı controller'ın diğer action metotlarını inşa ediyoruz şimdi.

ViewDatadaki bu köşeli parantez ve sözlük yapısı, viewbag ile dynamic nesneye dönüştürülmüştür. Dynamic nesneler, çalışma zamanında yazdığınız özelliklere o anda erişip sonuç almayı amaçlar.

```
HomeController.cs  + X
ers  denemesekizinciders.Controllers.HomeControl

}
0 başvuru
public ActionResult about()
{
    ViewBag.text1 = "Fatma Akalın";
    ViewBag.check1 = true;

    return View();
}
0 başvuru
public ActionResult deneme()
{
    ViewBag.text1 = "Fatma Akalın";
    ViewBag.check1 = true;
    ViewBag.list1 =
        new SelectListItem[]
        {
            new SelectListItem() { Text = "Elma" },
            new SelectListItem() { Text = "Armut" },
            new SelectListItem() { Text = "Çilek" }
        };

    return View();
}
```

Görünüm Ekle



Görünüm adı:

about

Şablon:

Empty (model olmadan)



Model sınıfı:



Seçenekler:

☐ Kısmi görünüm olarak oluştur

☐ Başvuru betik kitaplıkları

☐ Bir düzen sayfası kullanın:

(Bir Razor _viewstart dosyasında ayarlandıysa, boş bırakın)

Görünüm Ekle



Görünüm adı:

deneme

Şablon:

Empty (model olmadan)



Model sınıfı:



Seçenekler:

☐ Kısmi görünüm olarak oluştur

☐ Başvuru betik kitaplıkları

☐ Bir düzen sayfası kullanın:



(Bir Razor _viewstart dosyasında ayarlandıysa, boş bırakın)

Ekle

İptal

Deneme ve about viewlarının ekran çıktıları aşağıda verilmiştir.

```
html HomeController.cs deneme.cshtml about.cshtml x
```

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>about</title>
</head>
<body>
    <div>
        <b>Ad - Soyad :</b> @Html.TextBox("text1")
        <br />
        <b>Bool veri türü : </b> @Html.CheckBox("check1")
    </div>
    <div>
        <b>Ad- Soyad</b> @ViewBag.text1
        <br />
        <b>Bool veri türü</b> @ViewBag.check1
    </div>
</body>
</html>
```

about

localhost:44371/Home/about

PAMUKKALE ÜNİVE... Gmail YouTube Tumor

Ad - Soyad : Fatma Akalın

Bool veri türü : ☒

Ad- Soyad Fatma Akalın

Bool veri türü True

```
deneme.cshtml x about.cshtml Index.cshtml HomeController.cs
```

```
1
2 @{
3     Layout = null;
4 }
5
6 <!DOCTYPE html>
7
8 <html>
9 <head>
10    <meta name="viewport" content="width=device-width" />
11    <title>deneme</title>
12 </head>
13 <body>
14    <div>
15        @Html.DropDownList("list1")
16        <br />
17        @Html.DisplayText("text1")
18        <br />
19        @Html.Hidden("text1")
20        <br />
21        @Html.Value("text1")
22    </div>
23 </body>
24 </html>
25
```

deneme

localhost:44371/Home/deneme

PAMUKKALE ÜNİVE... Gmail YouTube Tumor

Elma

Fatma Akalın

Fatma Akalın

TempData

ViewBag, Viewdataya kıyasla daha kullanıcı dostu olan tempdata yapısı bir sonraki actionda verileri erişilebilir halde saklama imkanı sağlamaktadır. Yani sizin bir sayfa ileri gitmek hedefiniz için veriyi hafızada tutmayı sağlayan bir özellik sunuyor.

ViewData ve ViewBag ise başka bir sayfa açılışında dataları siler sadece o anlık kullanır.

Örnek vermek gerekirse TempData şu an Index.cshtml sayfasında olsun ve bu index sayfasını çalıştıran HomeControllerdaki actionumuzdur şeklinde bir varsayım yapalım. Index.cshtml sayfasından başka bir sayfaya tıkladığımızda başka bir actionumuz çalışacaktı. Ama o başka actionumuz içinde da TempData'da tuttuğumuz veri saklanacak. Yani TempData bu duruma izin verirken ViewData ve ViewBag ise başka bir sayfa açılışında verileri silmektedir.

Aynı controller içinde yeni bir action metodu oluşturmak ile başlayalım ve ardından görünüm/view ekleyip view sayfasında çağıralım.

0 başvuru

```
public ActionResult deneme2()
{
    TempData["text1"] = "Fatma Akalın";
    TempData["check1"] = true;
    return View();
}
```

Görünüm Ekle

Görünüm adı:

Şablon:

Model sınıfı:

Seçenekler:

☐ Kısmi görünüm olarak oluştur

☐ Başvuru betik kitaplıkları

☐ Bir düzen sayfası kullanın:

(Bir Razor _viewstart dosyasında ayarlandıysa, boş bırakın)

Ekle İptal

```
deneme2.cshtml HomeController.cs deneme.cshtml about.cshtml
1
2 @{
3     Layout = null;
4 }
5
6 <!DOCTYPE html>
7
8 <html>
9 <head>
10     <meta name="viewport" content="width=device-width" />
11     <title>deneme2</title>
12 </head>
13 <body>
14     <b>Ad - Soyad :</b> @TempData["text1"]
15     <br />
16     <b>Bool veri türü :</b> @TempData["check1"]
17 </body>
18 </html>
19
```

deneme2

localhost:44371/Home/deneme2

PAMUKKALE ÜNİVE... Gmail YouTube Tumor C

Ad - Soyad : Fatma Akalın
Bool veri türü : True

View'dan Controller'a Veri Gönderme

POST İşlemi

Az önce controllerlardan view e veri göndermiştik. Şimdi viewden controllera verileri nasıl alacağımızı öğrenelim. Örneğin, Kullanıcıya bir form sunduk ve bu form içinde kullanıcının yazdıklarını controllerda yakalayıp bunları database kaydetmek ya da başka işlemlerde kullanma amacımız var.

Bu doğrultuda HomeController ve view sayfası oluşturacağız. Burada homecontrollerda sayfayı çağırma bir get işlemidir. Kullanıcı bir sayfayı görmeden içine data yükleyip bize geri gönderemez. Dolayısıyla bir kullanıcının önce bir sayfayı çağırması gerekir. Bu nedenle ilk get işlemi yapılmalıdır.

Sayfa veri gönderdiğinde de post işlemi yapılıyor anlamına gelir. Bunun için [HttpPost] ifadesini metodun başına yazmamız gerekir.

[HttpGet]->BU, action sayfanın kullanıcı tarafından çağrıldığını söylerken

[HttpPost]->BU, sayfanın kullanıcı tarafından bize geri gönderildiğini söylüyor olacak. (Kullanıcı sayfaya veri yükleyip geri gönderdiğinde bu actionun çalışacağını söylüyoruz.)

Sayfa bize geri dönerken neleri alarak bize geri dönecek? Tüm sayfayı mı alıp geri döndüreceksin yoksa bir kısmını mı?

Önemli olan kısım text1 ve check1'in geri döndürülüyor olmasıdır. Bunun için using bloğu açılır ve bu using blogunun içerisine de Html BeginForm() oluşturulur. Bu yazılan kod using süslü parantezin başlangıç ve bitişine form elementi ekler.

BeginForm helper metodumuzda using kullanmamızın sebebi başlangıç bitiş noktalarını belli etmektir. Formumuz oluştuktan sonra içerisine html elementlerimizin oluşmasını ve datayı gönderecek olan buton elementimizi koyuyoruz.

Send butonuna tıkladığımızda formun içerisinde bulunan iki değer text1 ve check1 ismi ile geri döndürülecek. Dolayısıyla bunları nasıl yakalayacağım?

Bir uygulama ile öğrenelim...

Yeni bir proje açıp controllerımızı ekliyoruz

YALIN

ASP.NET Web Uygulaması (.NET Framework)

Proje adı

denemedokuzuncuders

Konum

D:\aspnet\1_ADESLER

Çözüm adı ⓘ

denemedokuzuncuders

☐ Çözümü ve projeyi aynı dizine yerleştir

Altyapı

.NET Framework 4.6.2

Proje "D:\aspnet\1_ADESLER\denemedokuzuncuders"

Yeni ASP.NET Web Uygulaması oluştur



Boş

ASP.NET uygulamaları oluşturmak için boş bir proje şablonu. Bu şablonda içerik yoktur.



Web Forms

ASP.NET Web Forms uygulamaları oluşturmak için bir proje şablonu. ASP.NET Web Forms, bilinen sürük ve bırak yöntemiyle olay denetimli modeli kullanarak dinamik web siteleri oluşturmanızı sağlar. Tasarım yüzeyi ile yüzlerce denetim ve bileşen, veri erişimi olan gelişmiş, güçlü, kullanıcı arabirimi denetimli web sitelerini hızlı bir şekilde oluşturmanızı sağlar.



MVC

ASP.NET MVC uygulamaları oluşturmaya yarayan bir proje şablonu. ASP.NET MVC, Model-View-Controller mimarisini kullanarak uygulamalar oluşturmanıza olanak sağlar. ASP.NET MVC en son standartları kullanan uygulamalar oluşturmak için hızlı, test güdümlü geliştirmeye olanak sağlayan birçok özellik içerir.



Web API

Tarayıcılar ve mobil cihazlar dahil olmak üzere çok çeşitli istemcilere erişebilen RESTful HTTP hizmetleri oluşturmaya yarayan bir proje şablonudur.



Tek Sayfalı Uygulama

ASP.NET Web API kullanarak JavaScript temelli zengin istemci tarafı HTML5 uygulamaları oluşturmak üzere kullanılan proje şablonudur. Tek Sayfalık Uygulamalar, HTML5, CSS3 ve JavaScript'in kullanıldığı istemci tarafı etkileşimler içeren zengin bir kullanıcı deneyimi sağlar.

Kimlik Doğrulama

Yok

Klasörleri ve çekirdek başvurularını ekle

- ☐ Web Forms
☒ MVC
☐ Web API'si

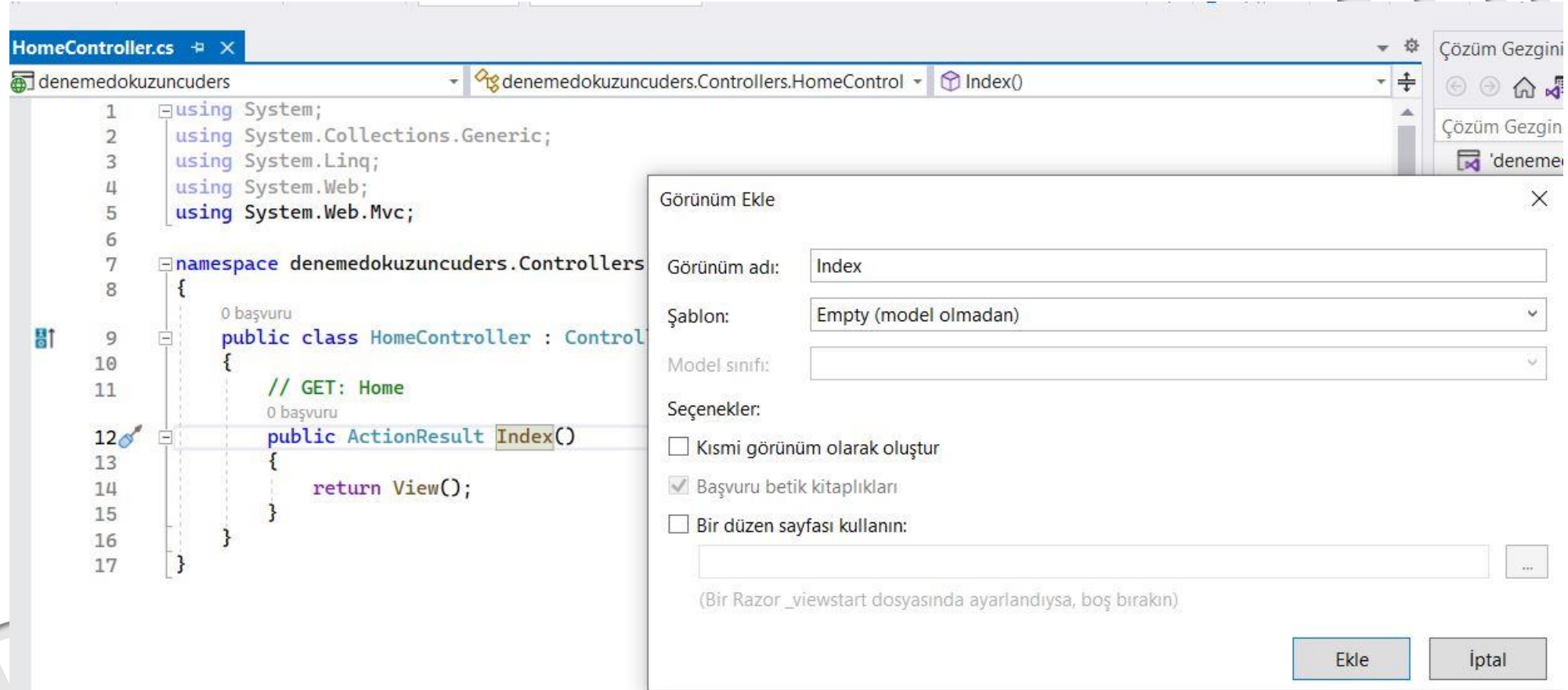
Gelişmiş

- ☒ HTTPS'yi Yapılandır
☐ Docker desteği
(Docker Desktop gerekiyor)

☐ Ayrıca birim testleri için bir proje oluştur

denemedokuzuncuders.Tests

Ardından view'ımızı ekliyoruz



The screenshot shows the Visual Studio IDE with the `HomeController.cs` file open. The code is as follows:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6
7 namespace denemedokuzuncuders.Controllers
8 {
9     // GET: Home
10     public class HomeController : Controller
11     {
12         // GET: Home
13         public ActionResult Index()
14         {
15             return View();
16         }
17     }
18 }
```

The `Index()` method is selected, and the `Görünüm Ekle` (Add View) dialog box is open. The dialog box contains the following fields and options:

- Görünüm adı:** Index
- Şablon:** Empty (model olmadan)
- Model sınıfı:**
- Seçenekler:**
 - ☐ Kısmi görünüm olarak oluştur
 - ☒ Başvuru betik kitaplıkları
 - ☐ Bir düzen sayfası kullanın:

At the bottom of the dialog box, there is a text field and a button labeled `...`. Below this, a note reads: `(Bir Razor _viewstart dosyasında ayarlandıysa, boş bırakın)`. At the bottom right, there are two buttons: `Ekle` (Add) and `İptal` (Cancel).

```
Index.cshtml  HomeController.cs
4      }
5
6      <!DOCTYPE html>
7
8      <html>
9      <head>
10         <meta name="viewport" content="width=device-width" />
11         <title>Index</title>
12     </head>
13     <body>
14         <div>
15             @{
16                 SelectListItem[] liste = new SelectListItem[3]
17                 {
18                     new SelectListItem() {Text="Fatma1", Value="a1"},
19                     new SelectListItem() {Text="Fatma2", Value="a2"},
20                     new SelectListItem() {Text="Fatma3", Value="a3"}
21                 };
22             }
23             @using (Html.BeginForm())
24             {
25                 @Html.TextBox("text1")<br />
26                 @Html.DropDownList("list1", liste)<br />
27                 @Html.CheckBox("check1") @("Onaylıyorum")
28                 <hr />
29                 <input type="submit" value="Send" />
30             }
31         </div>
32     </body>
33 </html>
34
35
```

```
Index.cshtml HomeController.cs x
denemedokuzuncuders denemedokuzuncuders.Controllers.HomeController Index(string text1, bo

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6
7 namespace denemedokuzuncuders.Controllers
8 {
9     public class HomeController : Controller
10     {
11         [HttpGet]
12         public ActionResult Index()
13         {
14             return View();
15         }
16         [HttpPost]
17         public ActionResult Index(string text1, bool check1, string list1)
18         {
19             //string list1 için bize value döner. textler ekranda gözüken değerler valuelar arkada tuttuğu kodlardır
20
21             var t1 = Request.Form["text1"];
22             var l1 = Request.Form["list1"];
23             var c1 = Request.Form.GetValues("check1")[0];
24             return View();
25         }
26     }
27 }
```

KAYNAKÇA

<https://learn.microsoft.com/tr-tr/aspnet/mvc/>

Veysel Uğur Kızmaz, ASP.NET MVC5, Kodlab Yayınları

<https://github.com/muratbaseren/udemy-yazilimcilarin-yukselisi-aspnet-my-evernote-sample>

Dr. Öğr. Üyesi Fatma AKALIN