1. ViewData veya ViewBag Kullanımı

ViewData veya ViewBag, basit veri iletimi için kullanılabilir.

Controller:

csharp
Kodu kopyala
```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        ViewBag.LayoutMessage = "Bu mesaj Layout'a gönderildi!";
        return View();
    }
}
```

Layout:

html
Kodu kopyala
```
<!DOCTYPE html>
<html>
<head>
    <title>@ViewBag.LayoutMessage</title>
</head>
<body>
    <h1>@ViewBag.LayoutMessage</h1>
    @RenderBody()
</body>
</html>
```

---

2. BaseController ile Veriyi Tüm Sayfalara Gönderme

Ortak bir veriyi tüm sayfalara göndermek için bir BaseController oluşturabilirsiniz.

BaseController:

csharp
Kodu kopyala
```
public class BaseController : Controller
{
    protected override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        ViewBag.SiteName = "MVC Projesi";
        base.OnActionExecuting(filterContext);
    }
}
```

Diğer Controller'lar:

```
public class HomeController : BaseController
{
```

```
    public ActionResult Index()
    {
        return View();
    }
}
```

Layout:

```
<h1>@ViewBag.SiteName</h1>
```

---

3. Partial View Kullanımı

Layout içinde Partial View çağırarak veriyi paylaşabilirsiniz.

Controller:

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        var user = new { Name = "Kullanıcı Adı", Role = "Admin" };
        return View(user);
    }

    public PartialViewResult UserDetails()
    {
        var user = new { Name = "Kullanıcı Adı", Role = "Admin" };
        return PartialView("_UserDetails", user);
    }
}
```

Partial View (_UserDetails.cshtml):

```
<p>Ad: @Model.Name</p>
<p>Rol: @Model.Role</p>
```

Layout:

```
<div>
    @Html.Action("UserDetails", "Home")
</div>
```

---

4. ViewModel Kullanımı

ViewModel ile Layout ve View arasında ortak bir veri yapısı tanımlayabilirsiniz.

ViewModel:

```
public class LayoutViewModel
{
    public string Title { get; set; }
```

```
    public string FooterMessage { get; set; }
}
```

Controller:

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        var model = new LayoutViewModel
        {
            Title = "Ana Sayfa",
            FooterMessage = "© 2024 Tüm Hakları Saklıdır."
        };
        return View(model);
    }
}
```

Layout:

```
<h1>@Model.Title</h1>
<footer>
    <p>@Model.FooterMessage</p>
</footer>
```

---

5. TempData Kullanımı

TempData, geçici veri saklamak için kullanılabilir. Bu yöntem genellikle bir yönlendirme (Redirect) sırasında veri taşımak için kullanılır.

Bu yöntemlerden hangisinin kullanılacağı, projenin ihtiyaçlarına ve mimari tercihlerinize bağlıdır. Daha büyük ve karmaşık projelerde genellikle ViewModel kullanımı önerilir.