



UPPSALA  
UNIVERSITET

# . NET-programmering

Razor

Kap 5



# Razor

- Inte ett programmeringsspråk
  - Det är ett server side markup language
- View Engine
  - Lägger till dynamiskt innehåll till html-filer
- Expressions
  - Läggas till i html tillsammans med C# kod
- Filändelsen
  - .cshtml



UPPSALA  
UNIVERSITET

# Exempel

`<ul>`

`@for (int i=0, i<10; i++)`

`{`

`<li>@i</li>`

`}`

`</ul>`



# Syntax

```
<html>
  <body>
    <!-- Single statement block -->
    @{ var myMessage = "Hello World"; }

    <!-- Inline expression or variable -->
    <p>The value of myMessage is: @myMessage</p>

    <!-- Multi-statement block -->
    @{
      var greeting = "Welcome to our site!";
      var weekDay = DateTime.Now.DayOfWeek;
      var greetingMessage = greeting +
        " Here in Huston it is: " + weekDay;
    }

    <p>The greeting is: @greetingMessage</p>
  </body>
</html>
```



UPPSALA  
UNIVERSITET

# Exempel

```
@{  
    var txt = "";  
    if(DateTime.Now.Hour > 12)  
    {  
        txt = "Good Evening";  
    }  
    else  
    {  
        txt = "Good Morning";  
    }  
}  
  
<html>  
    <body>  
        <p>The message is @txt</p>  
    </body>  
</html>
```



UPPSALA  
UNIVERSITET

# Mer Exempel

```
@{
    var totalMessage = "";
    if(IsPost)
    {
        var num1 = Request["text1"];
        var num2 = Request["text2"];
        var total = num1.AsInt() + num2.AsInt();
        totalMessage = "Total = " + total;
    }
}

<html>
<body>
    <form action="" method="post">
        <p><label for="text1">First Number:</label><br />
            <input type="text" name="text1" />
        </p>
        <p><label for="text2">Second Number:</label><br />
            <input type="text" name="text2" />
        </p>
        <p><input type="submit" value="Add" /></p>
    </form>
    <p>@totalMessage</p>
</body>
</html>
```



UPPSALA  
UNIVERSITET

# Mer Exempel

```
@{var price=50;}
```

```
<html>  
  <body>  
    @if (price>30)  
    {  
      <p>The price is too high.</p>  
    }  
  </body>  
</html>
```

# Koppling till .NET Core MVC

- C#-filer
  - using
  - using PartyInvites.Models;
- Razor View-filer
  - @model
  - @model PartyInvites.Models.GuestResponse;





UPPSALA  
UNIVERSITET

# Rensa PartyInvites



# Många import

- Skapa en View import fil
  - Högerklicka på Views-mappen
  - Välj add -> new item
  - Markera Web
  - Välj MVC View Imports Page
  - Behåll namnet `_ViewsImports.cshtml`
- Filen är tom
  - `@using PartyInvites.Models`
- I `.cshtml`-filen ändra till
  - `@model GuestResponse`



# Layout

- Skapa mallar med gemensamt utseende
  - Filer som ska delas av alla vyer läggs i mappen Shared som ska ligga i mappen Views
  - Även här börjar filnamnen med \_
  - Filnamnet bör innehålla ordet Layout, ex. `_BasicLayout.cshtml`

# Koppla innehåll till Layout

- `@RenderBody()`
  - Razor metod för att hämta innehållet från view-sidan som ska visas
- `@ViewBag.Title`
  - ViewBag tillåter data att skickas runt i en applikation



UPPSALA  
UNIVERSITET

# Koppla vy till layout

```
@{
```

```
    Layout = null;
```

```
}
```

```
@{
```

```
    Layout = "_BasicLayout";
```

```
    ViewBag.Title = "RsvpForm";
```

```
}
```



# View Startfil

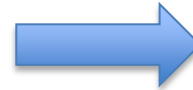
- `_ViewStart.cshtml`
  - Den fil som MVC tittar efter om man inte skriver något i vy-filerna om layout.
- Ska ligga i Views
  - Add new item, välj Razor View Start Page, behåll namnet

```
@{  
    Layout = "_Layout";  
}
```



# Byta Layout

```
@{  
    Layout = "_BasicLayout";  
}
```



```
@{  
}
```

```
@{  
    Layout = null;  
}
```

```
@{  
    Layout = "_OtherLayout";  
}
```



# ViewBag

- Dynamisk typ-attribut av ControllerBase
- Hålla reda på temporär data
- Existerar endast i den aktuella http request
  - Försöker man vidarebefordra den så har värdet förändrats till null
- Är egentligen en wrapper runt ViewData





UPPSALA  
UNIVERSITET

# ViewData

- Fungerar på samma sätt som ViewBag
- Är en dictionary (key-value)

# Processen

- Börja med tomt blad
  - Model first
- Färdig statisk sajt
  - Bygg en liten del i taget
  - Förvandla statiska html-filer till vyer (.cshtml)
    - Bryt ut gemensam layout
    - Se till att allt fungerar
  - Lägg på kontroller och modellen