

# ASP.NET MVC

©Copyright

Mahmud Al Hakim

mahmud@dynamicos.se

<http://www.webbacademy.se>

## Innehåll

- Introduktion till MVC
- Controller
- Action-metoder
- Views
- Arbeta med Layout-sidor och sektioner
- Route konfiguration
- Models
- MVC-formulär
- Säkerhet och validering
- Skicka data till en textfil

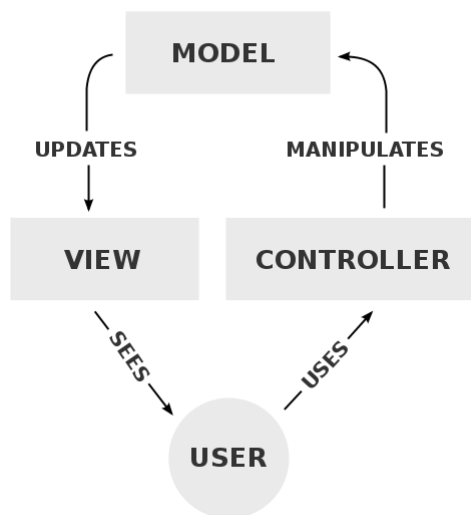
# Vad är MVC?

- MVC står för **Model-View-Controller**.
- MVC är ett Designmönster (design pattern) som används för att organisera komplexa applikationer.
- Introducerades 1979 tillsammans med programspråket Smalltalk.
- Med MVC delar man upp en applikation i separata lager.
  - **Model**: Data som ska bearbetas och hur detta är organiserat.
  - **View**: Användargränssnitt. Hur data ska presenteras för användaren.
  - **Controller**: Inputlogik och vilka operationer ska utföras på data.
- Separation mellan olika enheter inom en applikation kallas för **"Separation of Concerns"**.
- System ska bli lättare att designa, underhålla och förstå då man ser på dess olika delar som separata enheter med definierade ansvarsområden.

Copyright © 2015 Mahmud Al Hakim [www.webbacademy.se](http://www.webbacademy.se)

3

## MVC Interactions



Källa:  
<http://en.wikipedia.org/wiki/Model-view-controller>

4

## Vad är ASP.NET MVC?

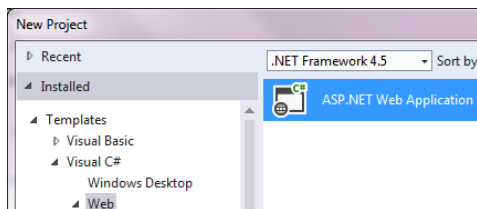
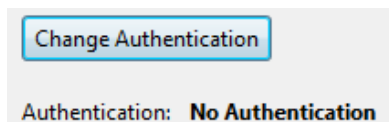
- ASP.NET MVC Framework är Microsofts MVC-ramverk för att skapa dynamiska webbsidor.
- Tekniken bygger på .NET och delar många funktioner med klassisk ASP.NET.
- Den första officiella versionen (ASP.NET MVC 1.0) släptes 2009.
- ASP.NET MVC gör det möjligt att skapa applikationer som kan testas och underhållas (Test-driven development TDD).
- Ger bättre kontroll över HTML som renderas.

Copyright © 2015 Mahmud Al Hakim [www.webbacademy.se](http://www.webbacademy.se)

5

## En första ASP.NET MVC Site

- Starta Visual Studio 2013
- Skapa ett nytt projekt av typen "ASP.NET Web Application". (finns under Visual C# - Web).
- Välj mallen "MVC"
- Välj "No Authentication".

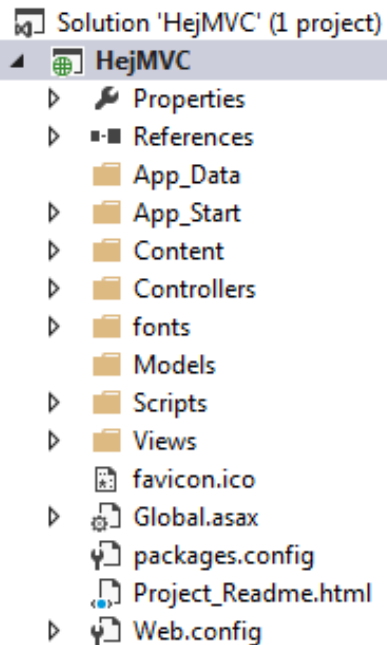


Copyright © 2015 Mahmud Al Hakim [www.webbacademy.se](http://www.webbacademy.se)

6

## Mapstruktur

- En typisk ASP.NET-MVC-applikation har följande mapstruktur.
- Vissa mappar är obligatoriska!
- **Tips**  
Behåll standardstrukturen. Ta inte bort eller ändra mapp- eller filnamn, om du inte 100% säker på vad du gör.



Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

7

## Controller

- Controller tar hand om användarens handlingar t.ex. en knapptryckning.
- Beroende på användarens val skickas info från en aktuell View till en Controller.
- Controller i sin tur skickar vidare info antingen till en Model eller en View.
- En controller är en cs-klass som innehåller ett antal metoder.
- Alla controller-klasser måste ärvas från klassen Controller och måste avslutas med ordet Controller t.ex. **HomeController.cs**

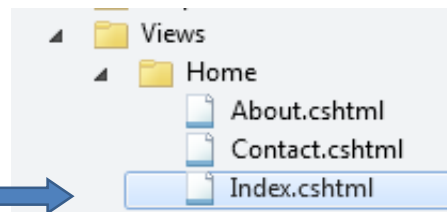
Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

8

## HomeController.cs

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }
    ...
}
```

En "Action-metod" returnerar en View som heter Index



Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

9

## Index.cshtml (View)

```
@{
    ViewBag.Title = "Välkommen till MVC";
}

<div class="jumbotron">
    <h1>ASP.NET</h1>
    <p class="lead">
        ASP.NET is a free web framework for building
        great Web sites and Web applications using
        HTML, CSS and JavaScript.
    </p>
</div>

. . .
```

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

10

# ViewBag

- ViewBag är en dynamisk "Data Dictionary".
- Både Action-metoden och dess motsvarande View har tillgång till detta objekt.

```
public ActionResult About()
{
    ViewBag.Message = "Välkomemn till MVC";
    return View();
}
```

# Skapa en ny ViewBag

## Controller

```
public ActionResult Index()
{
    ViewBag.Hej =
    DateTime.Now.Hour < 12 ? "God morgon" : "God dag";
    return View();
}
```

## View

```
@{
    ViewBag.Title = "Hem";
}

<h2>@ViewBag.Hej</h2>
```

## About.cshtml (View)

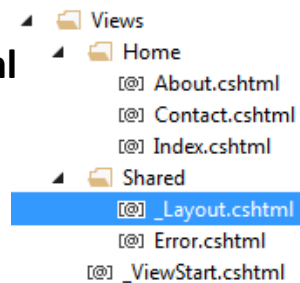
```
@{
    ViewBag.Title = "About";
}
<h2>@ViewBag.Title</h2>
<h3>@ViewBag.Message</h3>
```

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

13

## Layout (Template)

- I mappen Views -> Shared finns en mall som heter **\_Layout.cshtml**
- Innehållet i filen **\_ViewStart.cshtml** skickas till alla vyer automatiskt.
- I denna fil finns en Layout-direktiv som sätter sökvägen till layout-filen.



```
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
```

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

14

## Skapa en Partial View

**Add View**

View name:

Template:

Model class:

Options:

Create as a partial view

Reference script libraries

Use a layout page:

...

(Leave empty if it is set in a Razor \_viewstart file)

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

15

## \_Meny.cshtml (En Partial View)

Filen innehåller enbart en meny (inget annat).

```
<ul class="nav navbar-nav">
  <li>
    @Html.ActionLink("Home", "Index")
  </li>
  <li>
    @Html.ActionLink("About", "About", "Home")
  </li>
  <li>
    @Html.ActionLink("Contact", "Contact", "Home")
  </li>
</ul>
```

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

16



## Använd en Partial View i filen \_Layout.cshtml

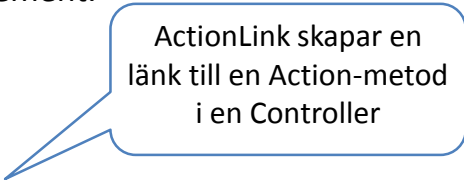
```
...  
<div class="navbar-collapse collapse">  
  
    @Html.Partial("~/Views/Shared/_Meny.cshtml")  
  
</div>  
...
```

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

17

## HTML Helpers

- MVC innehåller ett antal helpers till de mest vanliga typer av HTML-element.
- Exempel



ActionLink skapar en  
länk till en Action-metod  
i en Controller

```
@Html.ActionLink("Om oss", "About" , "Home")
```

Skapar följande länk

```
<a href="/Home/About">Om oss</a>
```

**Home** är namnet på en Controller och **About** är namnet på en Action-metod

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

18

## Sektioner

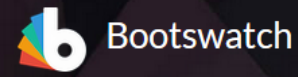
(Lägg till några sektioner i Layout-sidan)

```
. . .  
@RenderBody()  
@RenderSection("Annons1", false)  
@RenderSection("Annons2", false)  
  
<hr>  
<footer>  
  <p>&copy; @DateTime.Now.Year</p>  
</footer>  
. . .
```

## Använd valfria sektioner på valfria sidor (Views)

```
@section Annons1{  
<div>  
  <h1>Här kommer annons 1</h1>  
</div>  
}  
@section Annons2{  
<div>  
  <h1>Här kommer annons 2</h1>  
</div>  
}
```

## Tips! Ändra CSS-Mallen



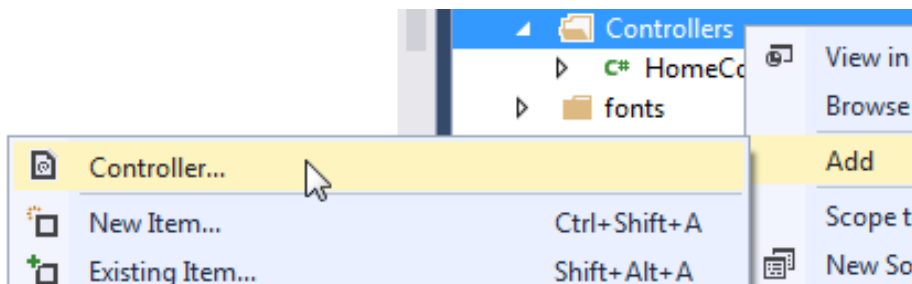
- Gå till
- <https://bootswatch.com/>
- Klicka på Themes
- Välj en CSS-Mall och klicka på Download
- Ladda ner **bootstrap.css**
- Spara i mappen Content.
- Ersätt den gamla!

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

21

## Skapa en ny Controller

- Högerklicka på mappen Controllers
- Välj Add -> Controller...
- Skapa en tom Controller (Empty)



Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

22

## Skapa en ny Action-metod

```
public class HejController : Controller
{
    public ActionResult Index()
    {
        return View();
    }

    public string Hello()
    {
        return "Hej min första Controller";
    }
}
```

Hur ser URLen ut för metoden Hello()?



Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

23

## En Action-metod med GET-Parametrar

```
public string Hello(string namn, string age)
{
    return "Hej " + namn
        + ". Din ålder: " + age ;
}
```



Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

24

## HtmlEncode

Tips. Använd metoden `HtmlEncode` i klassen `HttpUtility` för att skydda applikationen från skadlig kod.

Då skickas inga HTML-taggar eller andra teckenkoder.

Exempel:

```
public string Hello(string data)
{
    return HttpUtility.HtmlEncode(data);
}
```

Testkör,  
skicka  
specialtecken  
(t.ex. ÅÄÖ)  
Se källkoden!

## Global.asax

- Filen `Global.asax` innehåller metoden **`Application_Start`** som startas automatiskt när applikationen startas.
- Metoden anropar ett antal inställningsmetoder "Configuration Methods"  
Bl.a.

```
RouteConfig.RegisterRoutes(RouteTable.Routes);
```

## RouteConfig

Finns i mappen  
App\_Start

- Klassen RouteConfig har metoden RegisterRoutes som tar emot en referens till ett RouteCollection-objekt.
- Med hjälp av detta objekt sätts bl.a. en MapRoute

```
routes.MapRoute(  
    name: "Default",  
    url: "{controller}/{action}/{id}",  
    defaults: new { controller = "Home",  
                    action = "Index",  
                    id = UrlParameter.Optional  
    }  
);
```

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

27

## Skapa en ny MapRoute

```
routes.MapRoute(  
    name: "Welcome",  
    url: "Welcome",  
    defaults: new {  
        Controller = "Hej",  
        Action = "Hello"  
    }  
);
```

**OBS!**  
En ny MapRoute  
måste ligga före  
"Default"

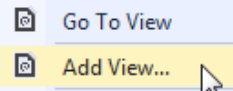


Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

28

## Skapa en ny View

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



Högerklicka på en  
Action-metod och  
välj  
Add View...

View\_name: Index

Template: Empty (without model)

Model class:

Options:

Create as a partial view

Reference script libraries

Use a layout page:

(Leave empty if it is set in a Razor \_viewstart file)

Add Cancel

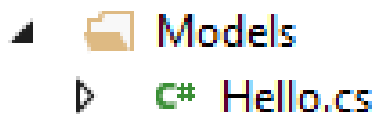
Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

29

## Skapa en ny Model

- Markera mappen Models och lägg till en ny klass.

```
namespace HejMVC.Models  
{  
    public class Hello  
    {  
        public string Hej  
        {  
            get  
            {  
                return DateTime.Now.Hour < 12 ? "God morgon"  
                : "God dag";  
            }  
        }  
    }  
}
```



Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

30

## Använd en Model i en Controller

```
using HejMVC.Models;

namespace HejMVC.Controllers
{
    public class HejController : Controller
    {
        public ActionResult Index()
        {
            var model = new Hello();
            return View(model);
        }
    }
}
```

Copyright © 2015 Mahmud Al Hakim [www.webbacademy.se](http://www.webbacademy.se)

31

## Skapa en View som använder en Model via en Controller

```
@model HejMVC.Models.Hello
```

```
<h2>@Model.Hej</h2>
```

Copyright © 2015 Mahmud Al Hakim [www.webbacademy.se](http://www.webbacademy.se)

32



## Skapa ett MVC-formulär med hjälp av HTML Helpers

```
@{ ViewBag.Title = "Kontakt"; }  
<h2>@ViewBag.Title.</h2>  
@using (Html.BeginForm())  
{  
    <label for="namn">Namn</label><br>  
    @Html.TextBox("namn");  
    <br>  
    <label for="epost">E-post</label><br>  
    @Html.TextBox("epost");  
    <br>  
    <label for="meddelande">Meddelande</label><br>  
    @Html.TextArea("meddelande");  
    <br>  
    <input type="submit" value="Skicka">  
}
```

Uppdatera filen  
Contact.cshtml  
(View)

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

33

## Data skickas tillbaka till Controller

```
[HttpGet]  
public ActionResult Contact()  
{  
    ViewBag.Message = "Your contact page.";  
    return View();  
}  
[HttpPost]  
public string Contact(string namn)  
{  
    return "Tack " + namn;  
}
```

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

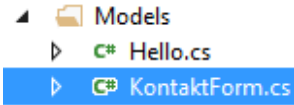
34

## Validera med hjälp av en Model Skapa en ny Model (KontaktForm.cs)

```
using System.ComponentModel.DataAnnotations;
namespace HejMVC.Models
{
    public class KontaktForm
    {
        [Required(ErrorMessage="Ange ditt namn!")]
        public string Namn { get; set; }

        [Required(ErrorMessage="Ange E-post!")]
        [EmailAddress(ErrorMessage="Ange en giltig E-postadress!")]
        public string Epost { get; set; }

        [Required(ErrorMessage="Skriv ett meddelande!")]
        public string Meddelande { get; set; }
    }
}
```



Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

35

## Uppdatera Controller

```
[HttpGet]
public ActionResult Contact()
{
    return View();
}
[HttpPost]
public ActionResult Contact(KontaktForm kontakt)
{
    if (!ModelState.IsValid)
    {
        return View(kontakt);
    }
    return RedirectToAction("Tack", "Home");
}
```

OBS!  
"Tack" är en  
Action-metod  
Som måste  
skapas

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

36

# Använd modellen i Contact.cshtml

```
@model HejMVC.Models.KontaktForm
@{ ViewBag.Title = "Kontakt"; }
<h2>@ViewBag.Title</h2>
@using (Html.BeginForm())
{
    <div>@Html.ValidationSummary()</div>
    <label for="namn">Namn</label><br>
    @Html.TextBox("namn");
    @Html.ValidationMessage("namn")
    <br>
    <label for="epost">E-post</label><br>
    @Html.TextBox("epost");
    @Html.ValidationMessage("epost")
    <br>
    <label for="meddelande">Meddelande</label><br>
    @Html.TextArea("meddelande", new { @cols = "80", @rows = "4" });
    @Html.ValidationMessage("meddelande")
    <br>
    <input type="submit" value="Skicka">
}

```

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

37

# Action-metoden "Tack" som returnerar en View

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

```

Filen Tack.cshtml

```
@{
    ViewBag.Title = "Tack";
}
<h2>Tack</h2>

```

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

38

## Skicka data till en textfil

```
public ActionResult Contact(KontaktForm kontakt)
{
    if(!ModelState.IsValid) {
        return View(kontakt);
    }
    else {
        using (StreamWriter writer = new
            StreamWriter(Server.MapPath("~/App_Data/kontaktform.txt"), true))
        {
            writer.WriteLine(kontakt.Namn+" "+ kontakt.Epost
                + ";" + kontakt.Meddelande);
        }
    }
    return RedirectToAction("Tack", "Home");
}
```

Glöm inte att  
**using System.IO;**  
Och spara textfilen i  
mappen **App\_Data**  
för att skydda!

## Säkerhet

### Cross Site Request Forgery (CSRF)

- CSRF är en hacker attack. Kallas även "one-click attack" eller "Session Riding".
- Attacken går ut på att locka användare till att klicka på en länk, förutsatt att någon form av kakor-baserat system används och offret har en giltig autentiseringskaka.
- Då kommer användarens webbläsare att skicka en begäran tillsammans med autentiseringskakan till webbservern som i sin tur kommer att utföra begäran eftersom den tror att offret avsiktligt har skickat denna begäran!

## Lägg till metoden AntiForgeryToken() i formuläret

```
@using (Html.BeginForm())  
{  
    @Html.AntiForgeryToken()  
    . . .  
}
```

- Formuläret genererar en hemlig "token" och är användarspecifik.
- När formuläret postas kontrollerar servern om denna token är korrekt.

```
<form action="/Home/Contact" method="post"><input  
name="__RequestVerificationToken" type="hidden"  
value="82eOUbGHUdyFvzHTgj4uhze1s7rLs87aCA7NEMtC_Twr  
pU4wTf4Ca2NanJM2wPuSNZbke36PzRurirOWFdKAj-0S3Q1" />
```

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

41

## Lägg till attributet ValidateAntiForgeryToken i Action-metoden som använder Post

```
[HttpPost]  
[ValidateAntiForgeryToken]  
public ActionResult Contact(KontaktForm kontakt)  
{  
    . . .  
}
```

Om man tar bort metoden  
ValidateAntiForgeryToken() från formuläret  
så går det inte att posta.  
(Testa gärna!)

*The required anti-forgery cookie "\_\_RequestVerificationToken" is not present.*

Copyright © 2015 Mahmud Al Hakim www.webbacademy.se

42