



MVC med Javascript och Ajax

Filip Ekberg



MVC med Javascript och Ajax

Lektion 7 – ASP.NET MVC

mail@filipekberg.se



Veckans mål

- **ASP.NET MVC Repetition & Diskussion**
 - Vad är MVC? Vad är ASP.NET MVC?
 - Hur skapar vi ett projekt med bra struktur?
 - När lägger man till nya Controllers, Vyer och vilka Actions skall man ha?
 - Vilka olika sätt kan vi använda AJAX på?
 - Hur skiljer sig JavaScript från C# - Client vs Server
 - Säkerhet – Vad bör vi tänka på?
 - Routing
- **Mer avancerat**
 - Vad är ett Filter? Hur skapar vi egna?
 - Vad är OWASP
 - Vad är en webbserver? Hur hostar man ASP.NET MVC?

Vad ska vi kunna nu?

Vad ska vi lära oss?

I denna kurs lär sig den studerande hur man jobbar med den naturliga uppdelningen av **data, GUI och logik** i sina projekt.

Detta görs med **MVC** som är ett ramverk. MVC har mer eller mindre blivit en **standard** för avancerade .NET-webbsidor. Arbetssättet gör att koden blir mer **återanvändbar och kostnadseffektiv**.

Man delar upp projektet i de **logiska lagren** i **Modell, View och Controller**. Varje logiskt lager kan ersättas av en ny modul för att skapa en helt ny applikation.

Vad ska vi lära oss?

- Skapa projekt i ramverket ASP.NET MVC
- Projektstruktur i Visual Studio
- Model-View-Controller:
 - Lagerstruktur och concept
 - Designmönster & Arkitektur
- Integrering av databas och Entity Framework
- Lägga till och modifiera: Layout, Controllers, Razor Views
- JavaScript, Ajax
- Säkerhet & Infrastruktur

Mål

Målet med kursen är att studerande ska kunna skapa **återanvändbar** kod enligt **MVC-ramverket**. På detta sätt kan studerande skapa webbprojekt som kan växa och expandera på ett kontrollerat sätt.

Mål

Efter genomgången kurs ska den studerande ha kunskaper och färdigheter inom

- Ramverket ASP.NET MVC med dess komponenter och tekniker.
- JavaScript och Ajax för dynamiska responsiva hemsidor
- Integrering av MVC ramverket och Entity Framework ramverket.

För att få betyget **godkänt**

- Har erhållit lägst betyget godkänt på tentamen
- Har klarat av alla obligatoriska moment
- Kan förklara MVC-modellens lagerstruktur
- Ska kunna skapa MVC-applikationer enligt god programmeringssed
- Kan lägga till och modifiera: Layout, Controllers, Razor Views
- Kan skapa responsiva hemsidor i MVC
- Förstår hur JavaScript och ASP.NET MVC arbetar tillsammans
- Förstår kopplingen mellan DOM (Document Object Model) och Ajax
- Kan skriva enklare JavaScript som används i MVC-projekt

För att få betyget **väl godkänt**

- Uppnått kunskapskraven för betyget godkänt
- Har erhållit lägst betyget **väl godkänt** på tentamen
- Kan använda MVC-projekt obehindrat
- Kan göra korrekta ändringar i modellen obehindrat
- Kan använda DOM-modellen med Ajax-kod
- Har fördjupad förståelse för JavaScript

ASP.NET MVC

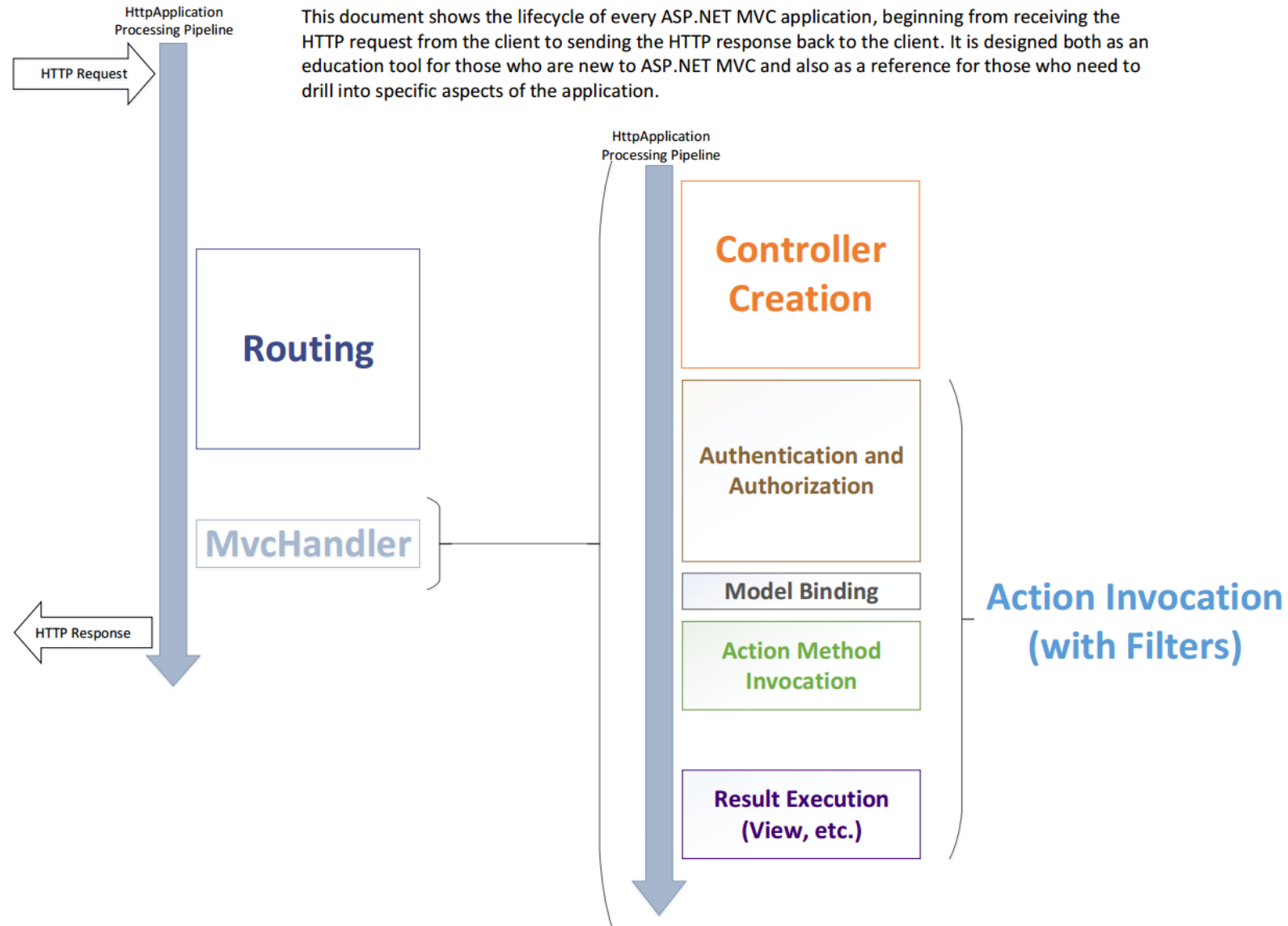
Vad är MVC?

Vad är ASP.NET MVC?

**När lägger man till en
ny Controller?**

Vad är skillnaden på JavaScript och C#?

ASP.NET MVC 5 APPLICATION LIFECYCLE – HIGH-LEVEL VIEW



Vad är en ViewEngine?

Och vad har ActionResult med det att göra?

**När produceras min
HTML?**

AJAX

Säkerhet


Varför HTTPS?

“Session Hijacking”

“Session Hijacking”


```
public ActionResult MyInvoices()
{
    var documents = new[] { "Filip_20130712.pdf",
        "Filip_20130713.pdf",
        "Filip_20130714.pdf"
    };

    return View(documents);
}
```



```
<h2>MyInvoices</h2>
<ul>
    @foreach (var document in Model)
    {
        <li><a href="/Home/DownloadDocument?document=@document">Download @document</a></li>
    }
</ul>
```

```
public ActionResult DownloadDocument(string document)
{
    var documentPath = Server.MapPath(Path.Combine("Invoices", document));

    if (!System.IO.File.Exists(documentPath)) { return null; }


    return File(documentPath, "application/pdf", document);
}
```

/Home/DownloadDocument?document=..\..\web.config

<http://www.filipekberg.se/2013/07/12/are-you-serving-files-insecurely-in-asp-net>

Routes

http://localhost:14845/Date



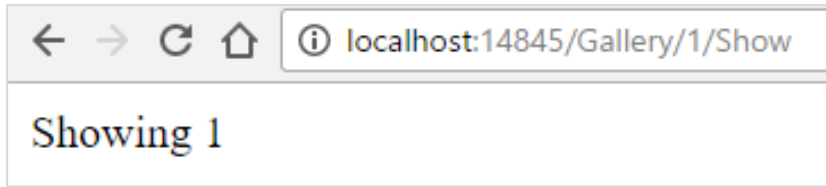
```
[Route("Date")]  
public ActionResult Date()  
{  
    return Content(DateTime.Now.ToLongTimeString());  
}
```

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapMvcAttributeRoutes();

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

```
routes.MapRoute(  
    name: "Example",  
    url: "mvc/example/{Controller}",  
    defaults: new { controller = "Home", action = "Index" }  
);
```



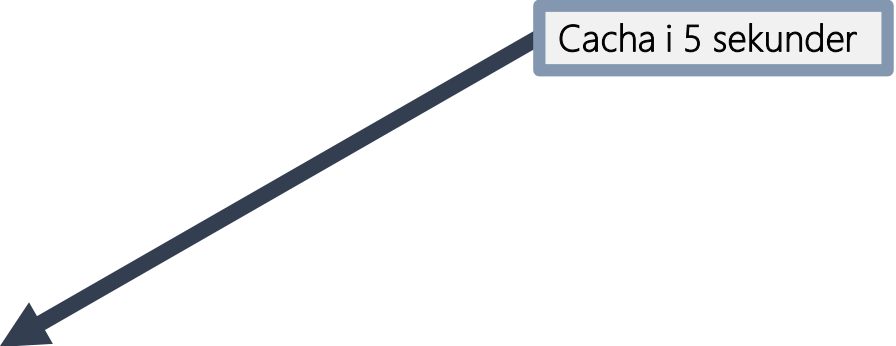
```
routes.MapRoute(  
    name: "GalleryRoute",  
    url: "Gallery/{id}/{action}",  
    defaults: new { controller = "Gallery", action = "Index" }  
);
```



```
public class GalleryController : Controller  
{  
    public ActionResult Show(int id)  
    {  
        return Content($"Showing {id}");  
    }  
}
```

Filters

Cacha i 5 sekunder

A dark blue arrow points from the callout box to the 'Duration = 5' part of the code.

```
[OutputCache(Duration = 5)]  
public ActionResult Date()  
{  
    return Content(DateTime.Now.ToLongTimeString());  
}
```

IAuthorizationFilter

IActionFilter

IResultFilter

ExceptionHandler

IAuthorizationFilter



IActionFilter



IResultFilter



IExceptionHandler

```
[CustomAuthorizationFilter]
[CustomActionFilter]
[CustomResultFilter]
[CustomExceptionFilter]
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }
}
```

```
public class AdministratorAuthorizationAttribute :
    FilterAttribute, IAuthorizationFilter
{
    public void OnAuthorization(AuthorizationContext filterContext)
    {
        var user = filterContext.HttpContext.User as ClaimsPrincipal;

        if (user == null)
        {
            filterContext.Result = new HttpUnauthorizedResult();

            return;
        }

        if (!user.HasClaim("Administrator", "true"))
        {
            filterContext.Result = new HttpUnauthorizedResult();

            return;
        }
    }
}
```

ASP.NET Identity

**Ett Identitetssystem som gör det
enkelt för oss att introducera ett
säkerhetslaget för inloggning**

**Stöd för inloggning med mer än
bara Användarnamn och
Lösenord – t.ex. Twitter,
Facebook, Google, etc.**

```
Install-Package Microsoft.Owin.Host.SystemWeb
```

```
Install-Package Microsoft.AspNet.Identity.OWIN
```

```
Install-Package Microsoft.AspNet.Identity.EntityFramework
```


```
public class MyIdentityDbContext
    : IdentityDbContext<IdentityUser>
{ }
```



```
var context = new MyIdentityDbContext();
var store = new UserStore<IdentityUser>(context);

userManager = new UserManager<IdentityUser>(store);
```

Hantera allt som har med
användaren att göra



Integrationstester & Automatiska UI Tester

Hur funkar ett integrationstest?

**Gör en riktig Request via en browser:
Automatiskt!**

Selenium

<http://www.seleniumhq.org/>

Körs som vanliga Test i Visual Studio






Install-Package Selenium.WebDriver

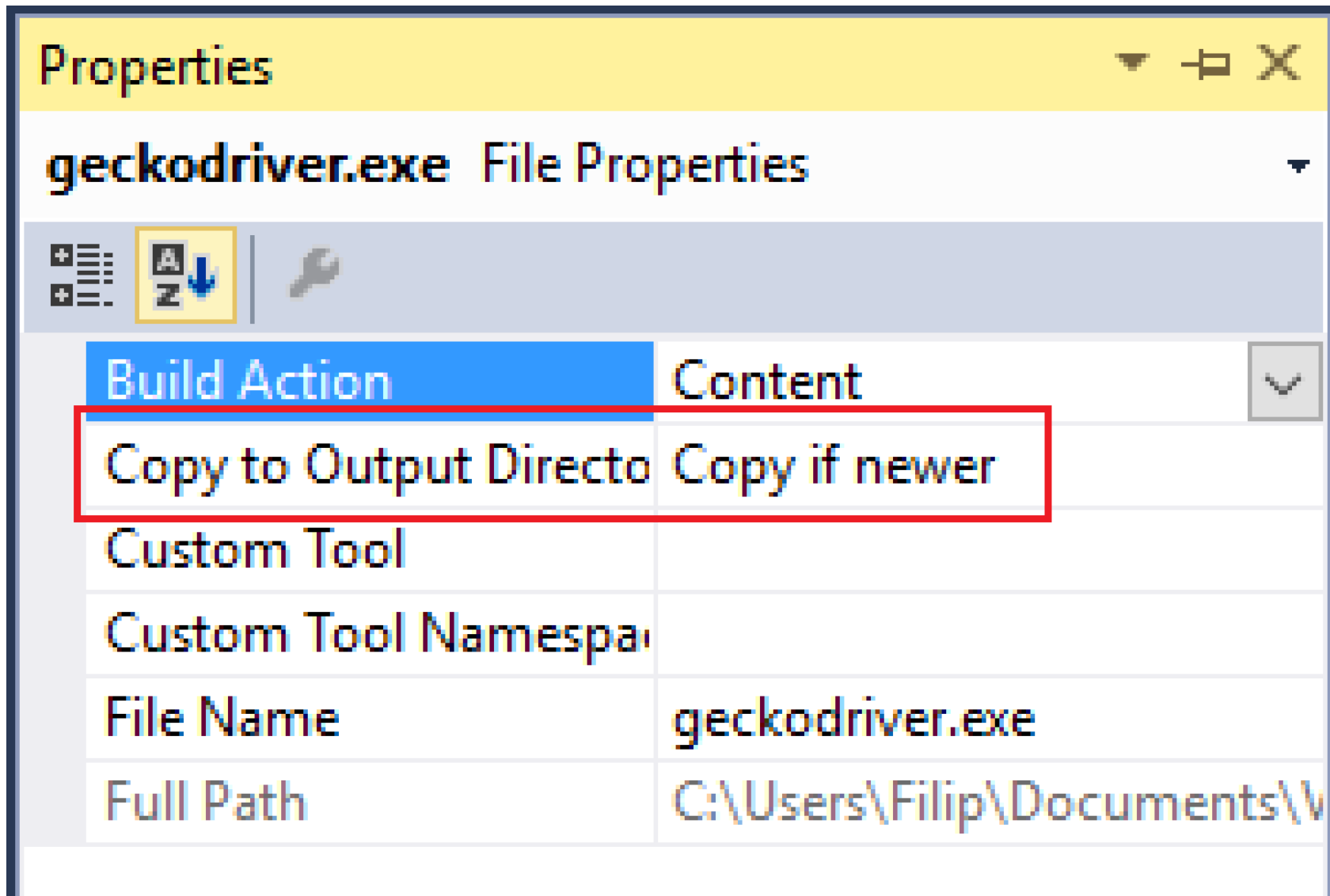
Ladda ner Firefox

Ladda ner GeckoDriver

<https://github.com/mozilla/geckodriver/releases>

MvcDemo.EntityFramework.Tests

- ▶  Properties
- ▶  References
- ▶  GalleryControllerTests.cs
-  geckodriver.exe
-  packages.config



```
var firefox = new FirefoxDriver();
```


```
[TestClass]
public class GalleryControllerTests
{
    [TestMethod]
    public void Index_ShouldReturn_IndexViewWithListOfImages()
    {
    }
}
```



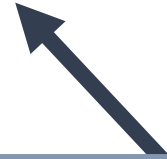
```
// ARRANGE
```

```
var firefox = new FirefoxDriver();  
firefox.Navigate().GoToUrl("http://localhost:58948/");
```

```
RemoveOldPhotos(firefox);
```



Vid integrationstester vill vi se till
att vi jobbar med ett "cleant" state



Vår ASP.NET MVC webbsida –
Detta kräver att vi startat
projektet!

```
// ACT
```

```
var fileUploadElement = firefox.FindElementByName("file");  
fileUploadElement.SendKeys(@"C:\Users\Filip\Desktop\sunset.jpg");
```

```
var name = Guid.NewGuid();
```

```
var nameElement = firefox.FindElementById("Name");  
nameElement.SendKeys(name.ToString());
```


```
var submitButton = firefox.FindElementById("submit");  
submitButton.Click();
```

```
Thread.Sleep(2000);
```


```
var firstTitle =  
    firefox.FindElementByCssSelector("div#result div.photo h3");
```



Peka på en fil som ska laddas upp



Skicka iväg vårt formulär



Vänta en stund, tills vi vet att vår
AJAX-request fått ett svar och vår
DOM har uppdaterats

```
// ASSERT
```

```
Assert.AreEqual(name.ToString(), firstTitle.Text);
```

```
firefox.Quit();
```

```
private static void RemoveOldPhotos(FirefoxDriver firefox)
{
    while (true)
    {
        try
        {
            var link = firefox.FindElementByCssSelector("div#result div.photo div a");
            if (link == null) break;

            link.Click();
            var alert = firefox.SwitchTo().Alert();
            alert.Accept();
        }
        catch
        {
            break;
        }
    }
}
```

Vår DOM uppdateras varje gång, så vi behöver leta efter nya referenser till alla länkar

Klicka på "OK" i Confirm-dialogen

Hittar vi ingen länk att trycka på kastas ett exception, då kan vi helt enkelt gå tillbaka till testet

**Vår ASP.NET MVC Applikation testat som
om en användare besöker den!**

När är detta bättre än enhetstester?

mail@filipekberg.se

