

# Dag 17 - mere UI

Brug af Layouts, Sections,  
Partial Views, Child Actions  
og Areas

# Emner

- Del 1: Modulær opbygning af User Interfaces (UI) i MVC
  - Brug af **layouts** til indhold som er delt af flere views
  - **Sections** til at arbejde med indhold som tilhører individuelle views
  - Brug af **partial views** til fragmenter af UI som kan "injectes" og genbruges i flere views som del elementer.
  - Indsættelse af dynamisk generet indhold i views med **child actions**
- Del 2: **Areas**

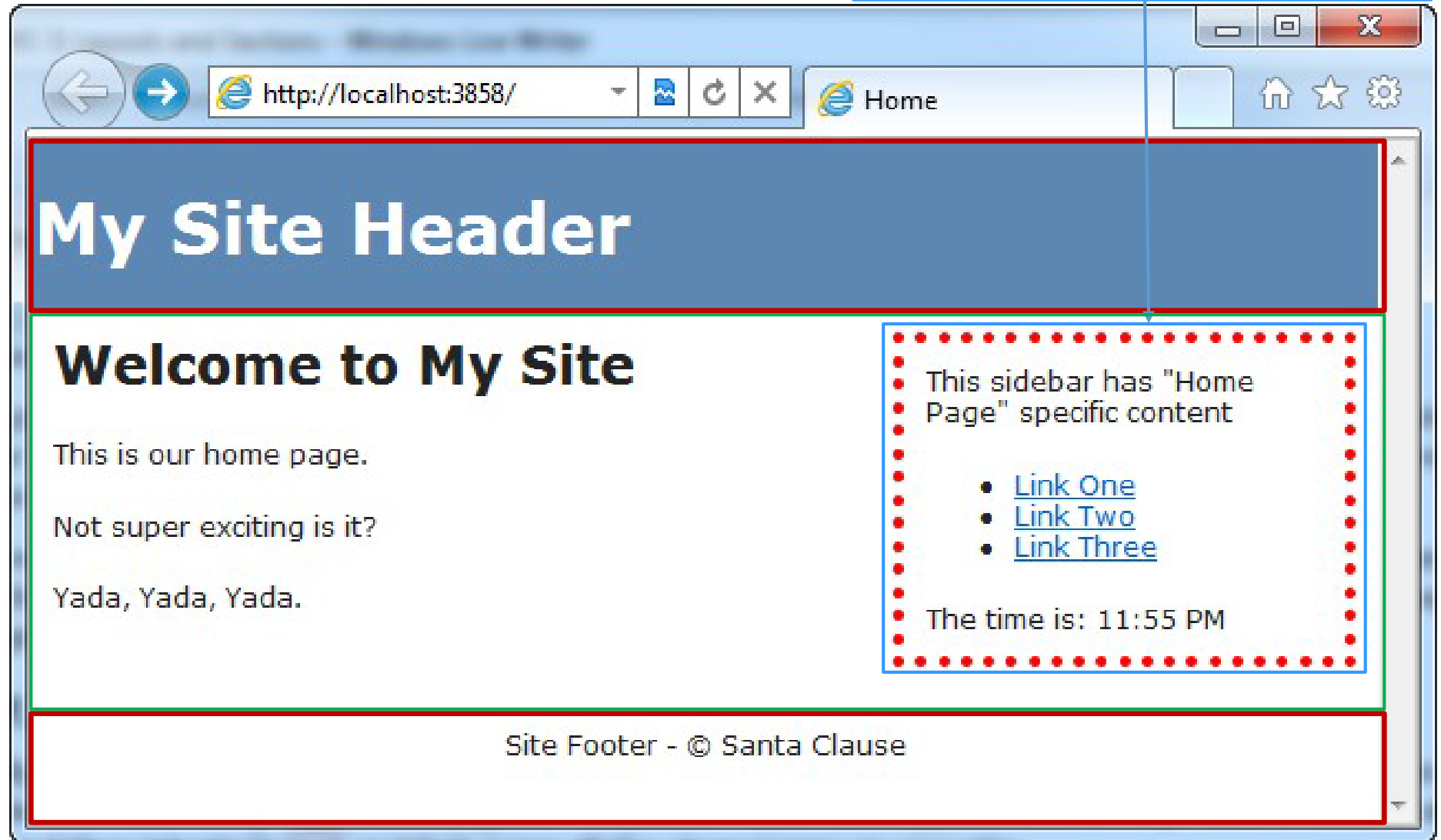
# Hurtigt overblik

Section / View

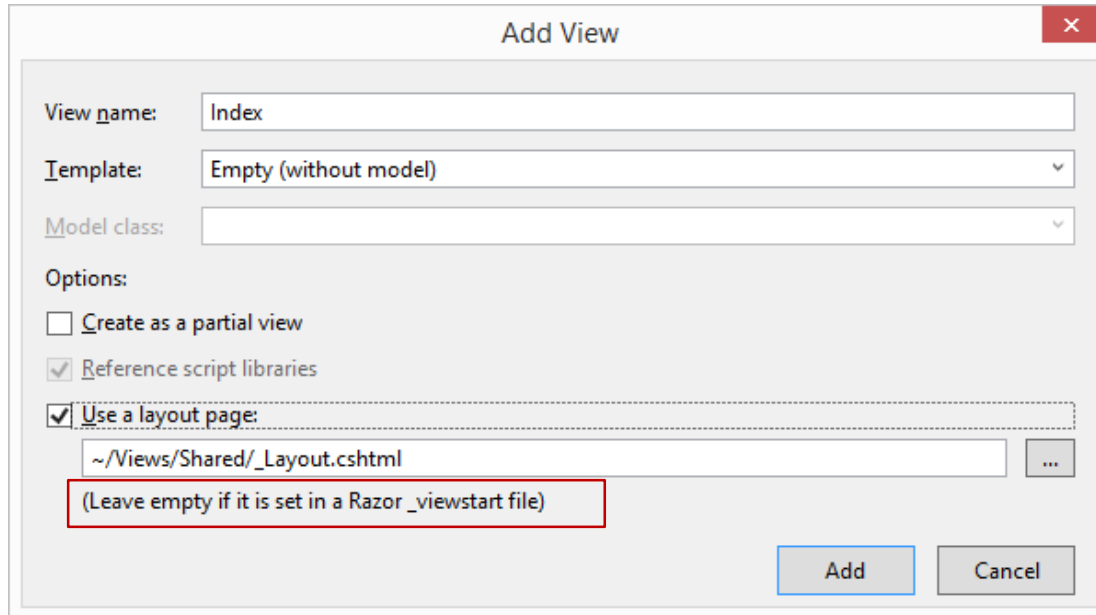
Layout fil

Layout fil

Child Action returnerer et Partial View



# Layout files – typisk til indhold som er fælles



View name: Index

Template: Empty (without model)

Model class:

Options:

- ☐ Create as a partial view
- ☒ Reference script libraries
- ☒ Use a layout page:

~/Views/Shared/\_Layout.cshtml

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

Add Cancel

default layout-fil  
angives i:  
Views\\_ViewStart.cshtml

```
1 @{  
2     Layout = "~/Views/Shared/_Layout.cshtml";  
3 }
```

Hvis Layout er defineret i Views\\_ViewStart.cshtml filen, så betyder det at ALLE views vil bruge dette Layout (med mindre at Layout eksplicit sættes til null i et view eller til en anden Layout fil).

# ~/Views/Shared/\_Layout.cshtml

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title> <!-- content from the view -->
</head>
<body>
    <div id="wrapper">
        <div class="layout">
            This is part of the layout file
        </div>

        @RenderBody() <!-- content from the view goes into this section -->

        <footer>[Insert footer information here]</footer>
    </div>
</body>
</html>
```

# Eksempel View fil, som bruger \_Layout fra før

```
@{
    ViewBag.Title = "Lesson 17 examples";
}
<!-- Goes into the Body-section of the layout file -->
<p>This is part of the view</p>
```

# Sections

- Lav section i view
- Brug section i Layout vha. RenderSection

# ~/Views/Shared/\_Layout.cshtml

## - Et mere avanceret eksempel

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>@ViewBag.Title</title>
  @RenderSection("Scripts", false)
</head>
<body>
  <div id="wrapper">
    // false means that the section is optional; true is default
    @RenderSection("Header", false)
    <div class="layout">
      This is part of the layout file
    </div>
    @RenderBody()
    <div class="layout">
      This is part of the layout file
    </div>
    @RenderSection("Footer", false)
    <footer>[Insert shared footer information here]</footer>
  </div>
</body>
</html>
```



# View filen (Bemærk @section)

- Med et `string[]` model fra controlleren

```
Index.cshtml  X
1  @model string[]
2  @{
3      ViewBag.Title = "Lesson 7, examples";
4
5  }
6  @section Header {
7      <div class="view">
8          @foreach (string str in new[] { "Index", "List", "Edit" }) {
9              @Html.ActionLink(str, str, null, new { style = "margin: 5px" })
10             }
11         </div>
12     }
13     <div class="view">
14         This is a list of fruit names:
15         @foreach (string name in Model) {
16             <span><b>@name</b></span>
17         }
18     </div>
19     @section Footer {
20         <div class="view">
21             This is the footer from the view
22         </div>
23     }
```

# Opgave

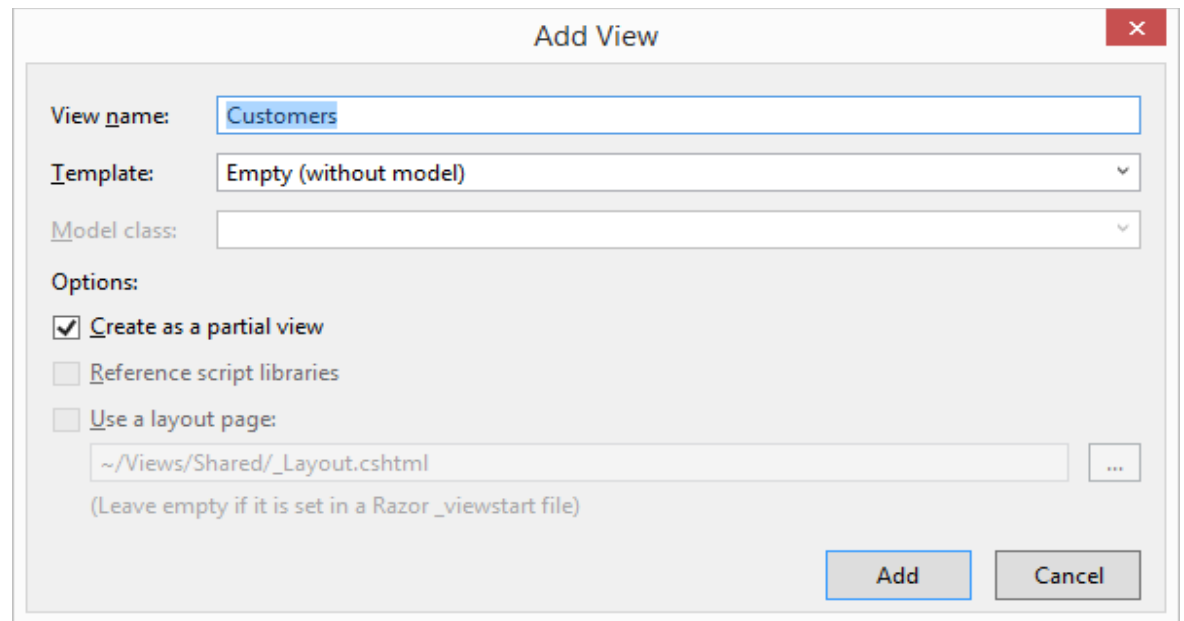
Lav opgave 17.1

# Overblik

- `Partial` (Indlejrer partial view i view)
- `RenderPartial`
- `Action` (Fuld programmeringsmodel i stedet for Razor)
- `RenderAction`
- `PartialView` (no layout page)
- Forskel på `Xxx` og `RenderXxx`:
  - `Xxx` returnerer værdien men sender den ikke i sig selv til output stream  
Dvs. man kan gemme værdien i en variabel og evt. ændre den inden den sendes til output  
Skal sendes til output som f.eks. `@Partial(...)`
  - `RenderXxx` sender værdien direkte til output stream  
Dvs. den sendes til output som den er. Bedre performance

# Partial Views

- Fragmenter af Razor (og data) **som skal inkluderes i andre view – de står ikke alene**
- Click checkboxen for at få et partial view:



The screenshot shows the 'Add View' dialog box. The 'View name' field contains 'Customers'. The 'Template' dropdown is set to 'Empty (without model)'. The 'Model class' dropdown is empty. Under the 'Options' section, the 'Create as a partial view' checkbox is checked. The 'Reference script libraries' checkbox is unchecked. The 'Use a layout page' checkbox is unchecked, and the text box below it contains '~/\_Views/Shared/\_Layout.cshtml'. At the bottom right are 'Add' and 'Cancel' buttons.

- Partial views skal være i `views\[controllernamefolder]` eller i `views\shared` folder (hvis det bruges af mere end en controller)

# Inklusion af Partial Views i andre views

- Du kan bruge præcis samme html og razor i et partial view som i et normalt view.

**Naming convention:** Partial og shared views starter med “\_”

- Du kan inkludere et partial view i **views og layout files**:

```
@Html.Partial("_PartialPage01")
```

- Vi kan også have strongly typed partial views, hvor en model kan sendes med

```
@Html.Partial("_PartialPage01", Model)
```

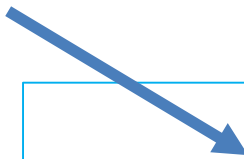
Et **strongly typed** partial view – som normalt

```
@model IEnumerable<string>
<div>
    Delicious fruit:
    <ul>
        @foreach (string name in Model) {
            <li>@name</li>
        }
    </ul>
</div>
```

# Child Actions – I Controlleren.

- **Child Actions** i forhold til normale Actions er som forholdet mellem partial views og normale views.
- **Child Actions kan IKKE kaldes direkte fra en url**, som normale Action metoder kan – f.eks. Home/MyChildAction (virker ikke!)
- **Child Actions** vil typisk bruges, hvis man har noget kode som man gerne vil bruge fra multiple controllers
  - Eksempel en data-driven widget
- Man kan bruge et Partial View til at vise output fra en **Child Action**

# Child Action Method – erklæres i en Controller



```
[ChildActionOnly]
public ActionResult ChildActionExample() {
    string[] myMovies = { "One Flew Over the Cuckoo's Nest",
                          "Schindler's List",
                          "Before Sunrise",
                          "Blue Ruin" };

    return PartialView(myMovies);
}
```

Her vil den så lede efter et Partial View som hedder ChildActionExample – altså ligesom normalt, hvor en Action metode med navn Index vil lede efter et View med Navn Index.



# Partial View til Child Action – her strongly typed

```
@model IEnumerable<string>

<p>
  Movies sent from a <strong>child action</strong>:
  <ul>
    @foreach (string movie in Model) {
      <li style="margin-left:30px;">@movie</li>
    }
  </ul>
</p>
```

# Kald af en Child Action fra et andet View:

Fra et View som hører til samme Controller :

```
@Html.Action("ChildActionExample")
```

Fra et View som hører til en anden Controller :

(så her kalder vi "MainMenu" Action metoden i "ToolsController")

```
@Html.Action("MainMenu", "Tools")
```

# Kald af en Child Action fra et andet View:

- I kaldet af child action kan man sende værdier med:
  - `@Html.Action("MainMenu", "Tools", new {P = <en eller anden værdi>})`

# Xxx vs RenderXxx

Eksempel: kald af Partial/RenderPartial:

- `@Html.Partial("VisEnPerson", Model)`
  - @ betyder, at det efterfølgende udtryk skal beregnes og sendes til output.
- `RenderPartiel:`
  - `@Html.RenderPartial("VisEnPerson", Model)` duer ikke, da `RenderPartiel` returnerer void.
  - Så pak det ind i script tags:
- ```
@{  
    Html.RenderPartial("VisEnPerson", Model);  
}
```

# Opgave

## Opgave 17.2

# Del 2: Areas

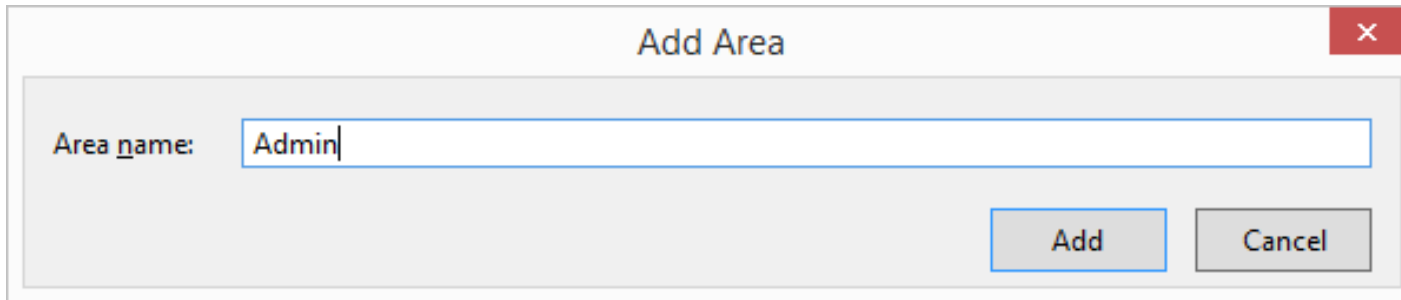
(og installation af css/js frameworks)

# Areas kan bruges til ....

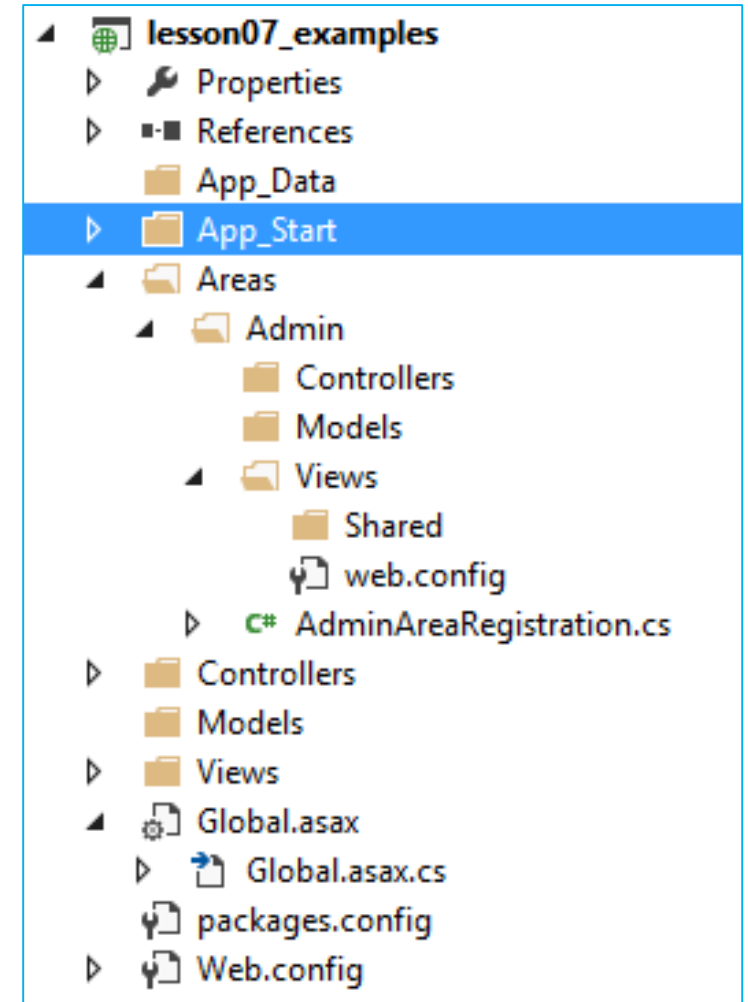
- Opdele et website i funktionelle **sub-segments** (som administration, billing, customer support etc.)
- Hvert Area kan have deres **egen folder struktur med controllers, models og views.**
- Areas er understøttet igennem **routing systemet**

# Tilføj et Area – højre klik og vælg :

- Add -> new scaffolded item -> MVC 5 Area ... første gang, ellers højre klik på Areas og add -> area



Vi kan så se at der sker nogle ændringer i nogle konfigurationsfiler – se næste slides:





# AdminAreaRegistration.cs

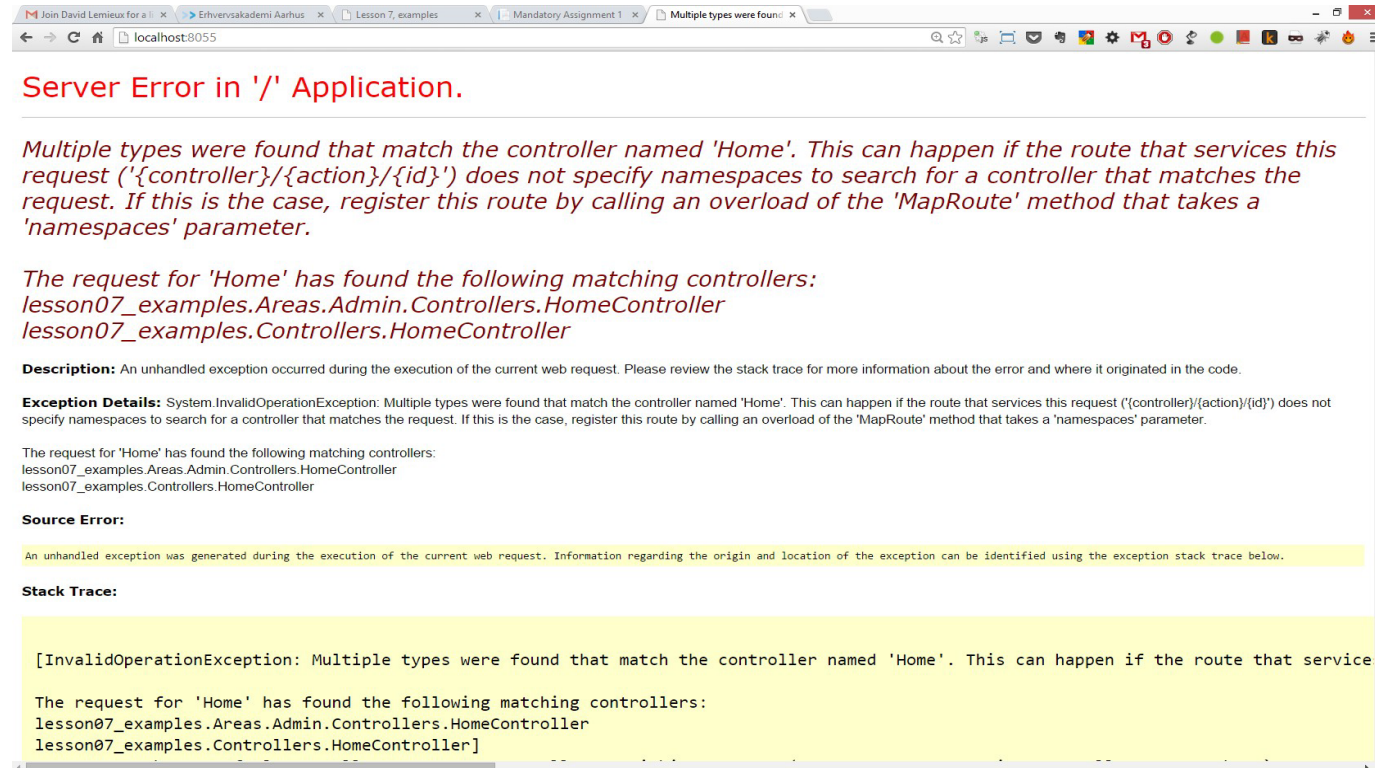
```
public class AdminAreaRegistration : AreaRegistration
{
    public override string AreaName
    {
        get
        {
            return "Admin";
        }
    }

    public override void RegisterArea(AreaRegistrationContext context)
    {
        context.MapRoute(
            "Admin_default",
            "Admin/{controller}/{action}/{id}",
            new { action = "Index", id = UrlParameter.Optional }
        );
    }
}
```

# Global.asax – registering Areas

```
public class MvcApplication : System.Web.HttpApplication
{
    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();
        RouteConfig.RegisterRoutes(RouteTable.Routes);
    }
}
```

# Pas på: multiple controllers med samme navn



I nyere versioner er dette vist ikke  
nødvendigvis et problem

# Løsninger

1. Brug forskellige navne – altså ikke 2 HomeController's, men **Admin**HomeController og HomeController
2. Tilføj et namespace til controller

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

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional },
        namespaces: new[] { "lesson07_examples.Controllers" }
    );
}
```

# Links fra et Area til et andet Area (i View)

```
@Html.ActionLink("Admin Only", "Index", "Home", new { area = "Admin" }, null)
```

**Vil så lave et link til Home/Index i Admin areaet. Sidste parameter som er null er til css som sædvanligt, hvis linket skal styles.**

# Installation af frontend/css framework i MVC

# Installation af jQuery

To install jQuery, run the following command in the Package Manager Console

```
PM> Install-Package jQuery
```

Packet manager console kan findes under "Tools->Nuget packet manager" menu i Visual Studio.

# Installation of Bootstrap (CSS framework)

To install Bootstrap, run the following command in the [Package Manager Console](#)

```
PM> Install-Package Twitter.Bootstrap
```



# Man også spørge efter bestemte versioner:

---

```
Install-Package jQuery -version 1.10.2  
Install-Package jQuery.Validation -version 1.11.1  
Install-Package Microsoft.jQuery.Unobtrusive.Validation -version 3.0.0  
Install-Package Bootstrap -version 3.0.0  
Install-Package Microsoft.jQuery.Unobtrusive.Ajax -version 3.0.0
```

---

# Brug af CSS and JS i Layout filen – brug drag/drop

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width" />
6      <title>@ViewBag.Title</title>
7      <link href="/Content/bootstrap.css" rel="stylesheet" />
8      <link href="/Content/bootstrap-theme.css" rel="stylesheet" />
9      <script src="/Scripts/bootstrap.js"></script>
10     <script src="/Scripts/jquery-1.9.0.js"></script>
11
12     @RenderSection("Scripts", false)
13
14
15 </head>
16 <body>
```