WEB SYSTEMS 202

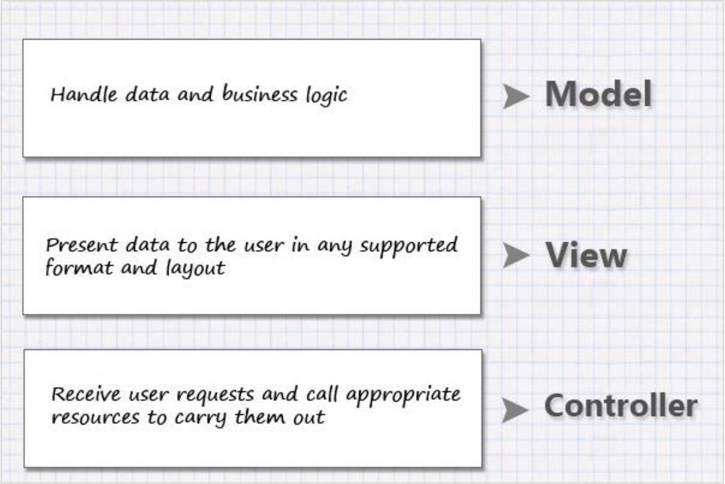
Marinda Taljaard
Office 09 02 29
marinda.taljaard@mandela.ac.za

Model-View-Controller Web Applications

DIFFERENT WAYS TO CREATE WEBSITES

- Semester 1: HTML and CSS pages
- Semester 2:
  - ASP.NET forms to build dynamic websites using drag-and-drop, event-driven model.
  - ASP.NET MVC
    - Powerful, patterns-based way to build dynamic websites
    - Enables clean separation of concerns

Model-View-Controller (MVC)



Handle data and business logic ➤ Model

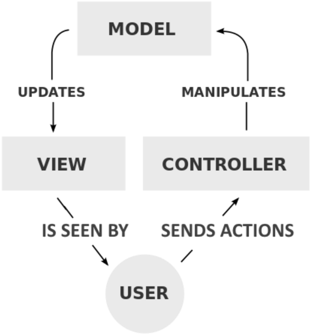
Present data to the user in any supported format and layout ➤ View

Receive user requests and call appropriate resources to carry them out ➤ Controller

- Three main components – different tasks

Model

- Data and rules applying to the data
- Model gives the controller a data representation of what user requested
  - Controller will determine what to do with it



MODEL

UPDATES

MANIPULATES

VIEW

CONTROLLER

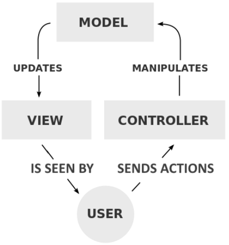
IS SEEN BY

SENDS ACTIONS

USER

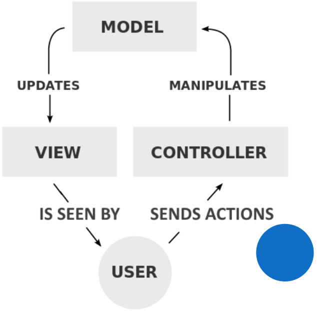
Controller

- Manages user requests (received as HTTP GET or POST requests)
- Main function is to call and coordinate necessary resources/objects needed to perform user action
- Issues commands to model
- Updates the view whenever data changes
- Keeps view and model separate



View

- Provides different ways to present the data received from the model
- May be templates were data is filled
- Could have several different views – controller will decide which to use
- The user interface (HTML/ASP controls) necessary to render the model to the user



Class exercise

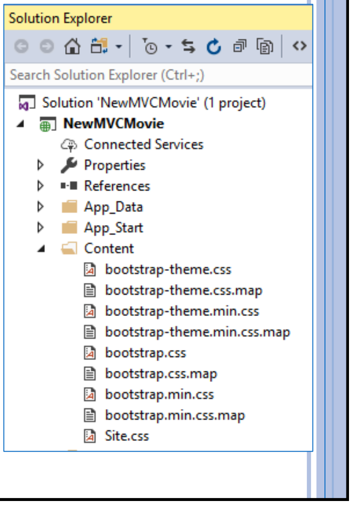
- Copy the NewMVCMovie folder to your H drive (from Courses folder)
  - Basic Movie Application with database
  - Instructions to create this available in *MVC Application Tutorial.pdf*
- Open the Project NewMVCMovie in Visual studio (there is a sln file; File - Open - Project)
- Run the project
  - Some menu items might not be functional yet
- Inspect the project – look at all the files created, to determine things like:
  - What types of files?
  - What goes into each type of file?
  - How is the syntax used different?
  - Where does the formatting instructions come from?

Discussion questions

- Consider the URL: http://localhost/NewMVCMovie/Home/Index - where does the last three components come from?
  - NewMVCMovie
  - Home
  - Index
- NewMVCMovie – Name of the project
- Home – Controller
- Index – Name of the method
  - Index method is the default method, if no method specified

DISCUSSION QUESTIONS

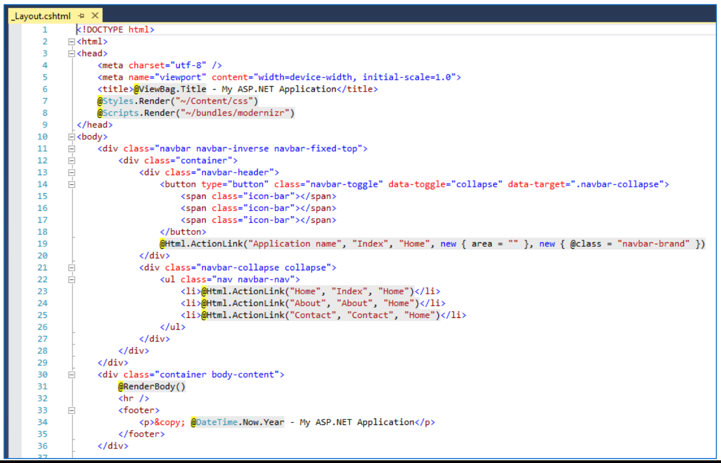
- A MVC application still uses css files
  - Where are they saved?
  - How are they accessed?
- You should investigate the matter of using bootstrap
  - Can you override formatting done using bootstrap?



```
Solution 'NewMVCMovie' (1 project)
  NewMVCMovie
    Connected Services
    Properties
    References
    App_Data
    App_Start
    Content
      bootstrap-theme.css
      bootstrap-theme.css.map
      bootstrap-theme.min.css
      bootstrap-theme.min.css.map
      bootstrap.css
      bootstrap.css.map
      bootstrap.min.css
      bootstrap.min.css.map
      Site.css
```

DISCUSSION QUESTIONS

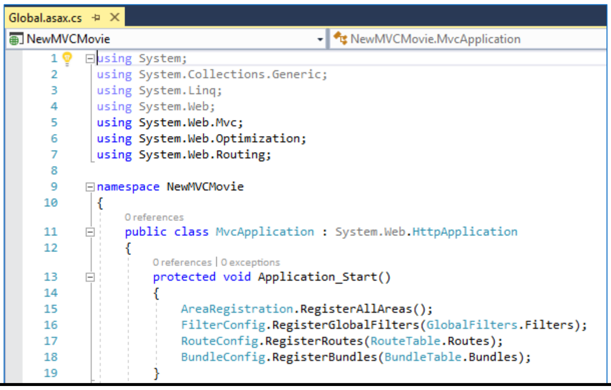
- Is there an equivalent to an ASP.net master page?



```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                @Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li>@Html.ActionLink("Home", "Index", "Home")</li>
                    <li>@Html.ActionLink("About", "About", "Home")</li>
                    <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
                </ul>
            </div>
        </div>
    </div>
    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
        </footer>
    </div>
```

Discussion Questions

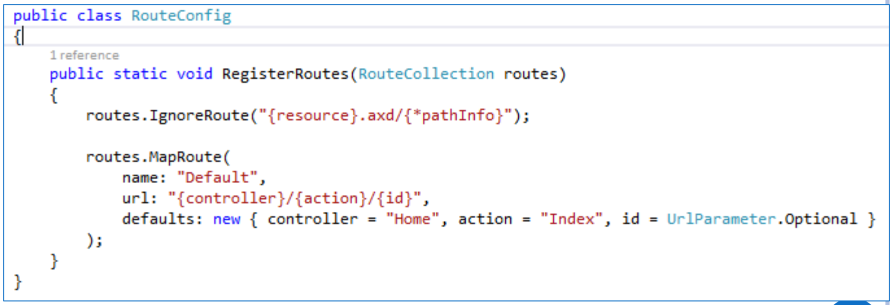
- Locate and open the Global.asax file
  - Application_Start() method
  - Calls made to various configuration methods to setup application’s working state



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace NewMVCMovie
{
    0 references
    public class MvcApplication : System.Web.HttpApplication
    {
        0 references | 0 exceptions
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
```

Discussion Questions

- Call to RegisterRoutes method of RouteConfig class
  - See App_Start folder
  - Default RouteConfig class:

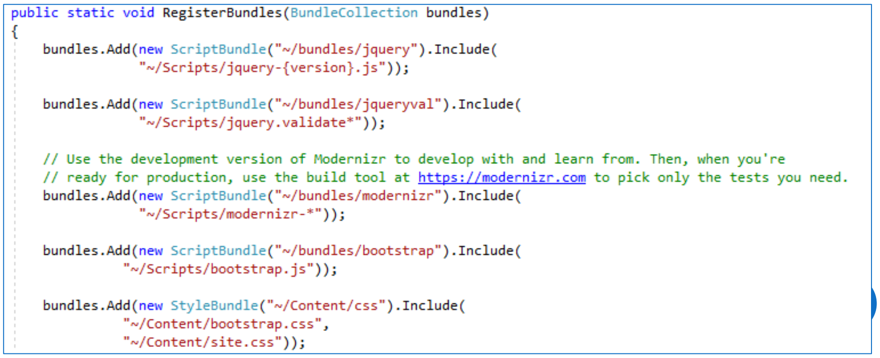


```
public class RouteConfig
{
    1 reference
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

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

  - Every MVC App needs at least 1 route definition (Default)

Discussion Questions

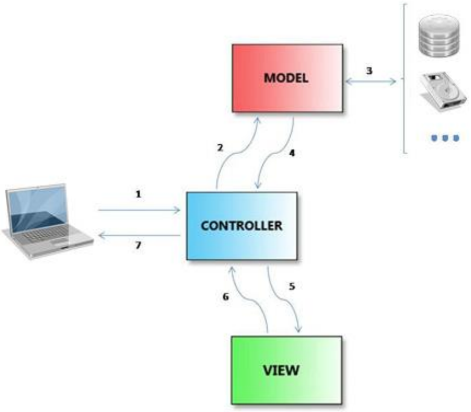
- Call to RegisterBundles method of BundleConfig class
  - App_Start folder
  - Reducing network traffic by bundling a number of files into a single bundle – single browser request to server



```
public static void RegisterBundles(BundleCollection bundles)
{
    bundles.Add(new ScriptBundle("~/bundles/jquery").Include(
                "~/Scripts/jquery-{version}.js"));

    bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
                "~/Scripts/jquery.validate*"));

    // Use the development version of Modernizr to develop with and learn from. Then, when you're
    // ready for production, use the build tool at https://modernizr.com to pick only the tests you need.
    bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(
                "~/Scripts/modernizr-*"));

    bundles.Add(new ScriptBundle("~/bundles/bootstrap").Include(
              "~/Scripts/bootstrap.js"));

    bundles.Add(new StyleBundle("~/Content/css").Include(
              "~/Content/bootstrap.css",
              "~/Content/site.css"));
```

Example: Movie app

- Simple movie listing application that supports creating, editing, searching and listing movies from a database



EXAMPLE: MOVIE APP

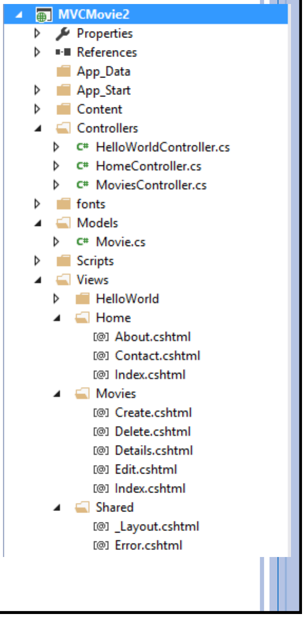
- Controller:
  - Receive user request
  - Examine request and call model asking for list of available movies
- Model
  - Get information from the database (or wherever stored)
  - Apply filters or logic and return data representing list of movies
- Controller:
  - Select appropriate view (pc, mobile, etc.)
- View
  - Present data to user

ADVANTAGES

- Support for different user using different devices
  - Interface different but model provides the same data
  - Controller will choose which view to use
- Separation of concern
  - Loosely coupled: should be able to change anything on any 1 of the 3 without affecting the others
- Reduces complexity when designing large applications
- Structured code making it easier to maintain, test and reuse

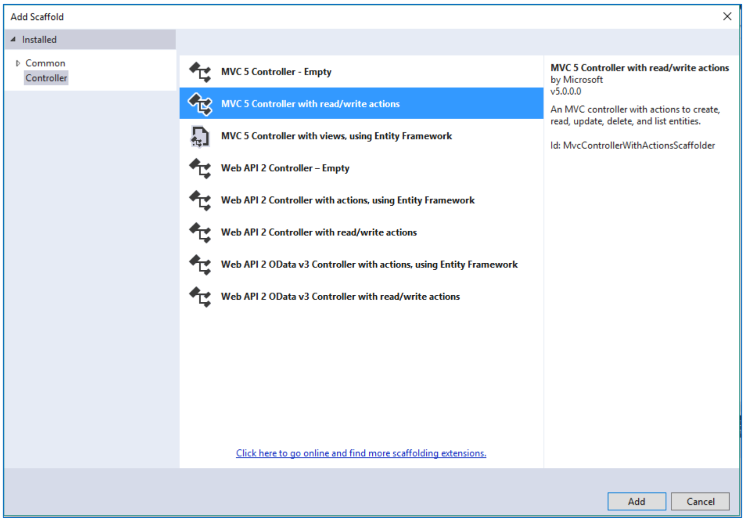
Model View Controller

- Naming conventions are important
  - HomeController linked to Home View
  - MoviesController linked to Movies View
- Each View object is linked to a Controller action method (e.g. when Create method is called the Create View object linked to the current controller will be used



```
◢ MVCMovie2
   ▷ Properties
   ▷ References
     App_Data
   ▷ App_Start
   ▷ Content
   ◢ Controllers
      ▷ C# HelloWorldController.cs
      ▷ C# HomeController.cs
      ▷ C# MoviesController.cs
   ▷ fonts
   ◢ Models
      ▷ C# Movie.cs
   ▷ Scripts
   ◢ Views
      ▷ HelloWorld
      ◢ Home
           [@] About.cshtml
           [@] Contact.cshtml
           [@] Index.cshtml
      ◢ Movies
           [@] Create.cshtml
           [@] Delete.cshtml
           [@] Details.cshtml
           [@] Edit.cshtml
           [@] Index.cshtml
      ◢ Shared
           [@] _Layout.cshtml
           [@] Error.cshtml
```

Add controller to access model's data



Add Scaffold

◢ Installed
- ▷ Common
  - Controller

- MVC 5 Controller - Empty
- MVC 5 Controller with read/write actions
- MVC 5 Controller with views, using Entity Framework
- Web API 2 Controller – Empty
- Web API 2 Controller with actions, using Entity Framework
- Web API 2 Controller with read/write actions
- Web API 2 OData v3 Controller with actions, using Entity Framework
- Web API 2 OData v3 Controller with read/write actions

MVC 5 Controller with read/write actions
by Microsoft
v5.0.0.0

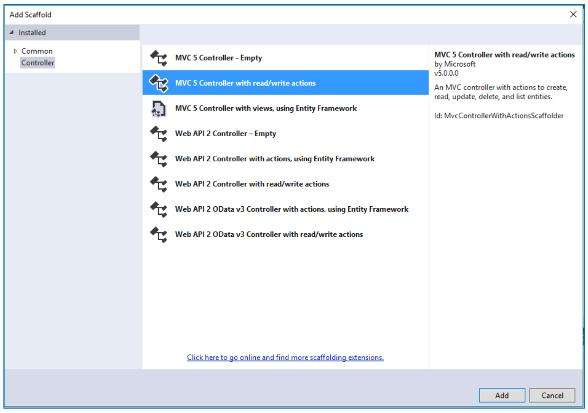
An MVC controller with actions to create, read, update, delete, and list entities.

Id: MvcControllerWithActionsScaffolder

Click here to go online and find more scaffolding extensions.

ADD CONTROLLER

- MVC 5 Controller – Empty
- MVC 5 Controller with read/write actions
  - Provides some skeleton methods to work with


Add Scaffold
Installed
Common
Controller
MVC 5 Controller - Empty
MVC 5 Controller with read/write actions
MVC 5 Controller with views, using Entity Framework
Web API 2 Controller – Empty
Web API 2 Controller with actions, using Entity Framework
Web API 2 Controller with read/write actions
Web API 2 OData v3 Controller with actions, using Entity Framework
Web API 2 OData v3 Controller with read/write actions
Click here to go online and find more scaffolding extensions.
Add
Cancel

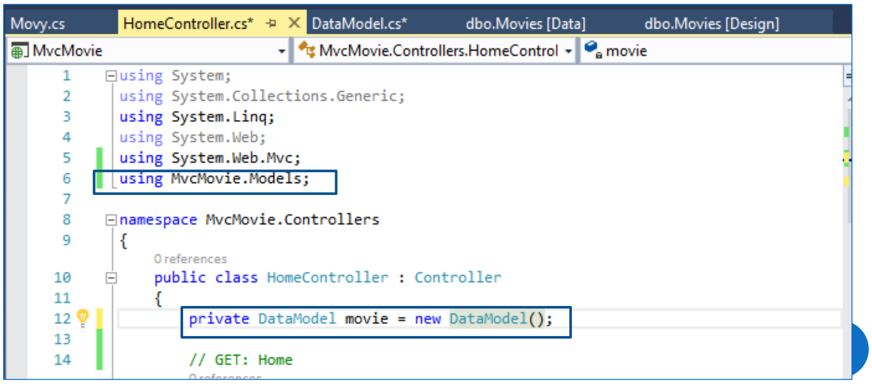
MVC 5 Controller with read/write actions
by Microsoft
v5.0.0.0

An MVC controller with actions to create, read, update, delete, and list entities.

Id: MvcControllerWithActionsScaffolder

EDIT CONTROLLER

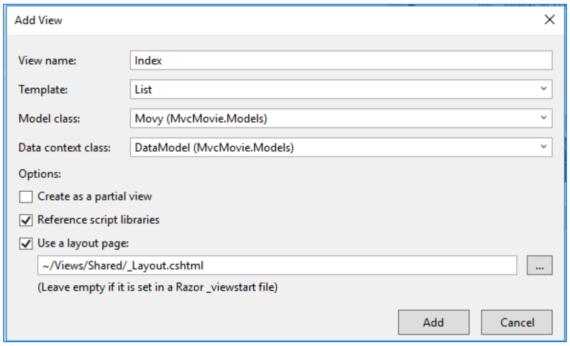
- Which Data Model to use (link to)
- Create an object to work with



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using MvcMovie.Models;

namespace MvcMovie.Controllers
{
    0 references
    public class HomeController : Controller
    {
        private DataModel movie = new DataModel();

        // GET: Home
```

Adding Views

- When a method is called the controller wants to direct the user to the relevant View
- Right-click on the method, Add view
  - Select relevant template, Model class, Data content class and Layout page



Add View

View name: Index

Template: List

Model class: Movy (MvcMovie.Models)

Data context class: DataModel (MvcMovie.Models)

Options:

☐ Create as a partial view

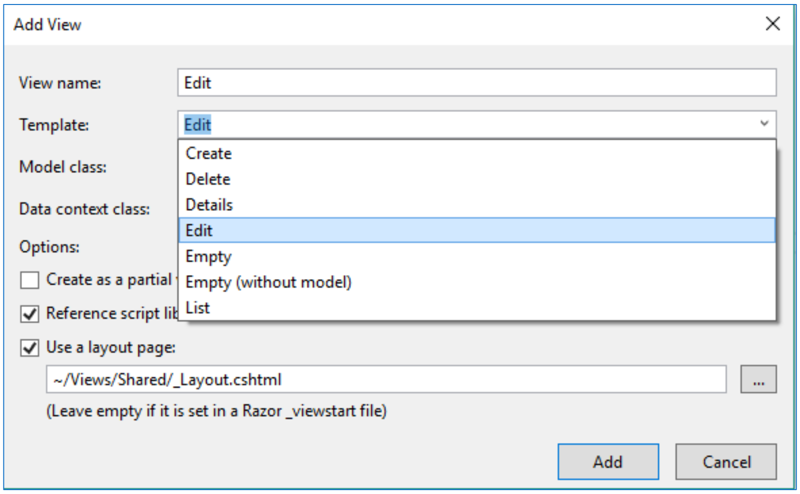
☑ Reference script libraries

☑ Use a layout page:

~/Views/Shared/_Layout.cshtml

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

Selecting View templates



Add View

View name: Edit

Template: Edit

Model class:

Data context class:

Options:

☐ Create as a partial

☑ Reference script lib

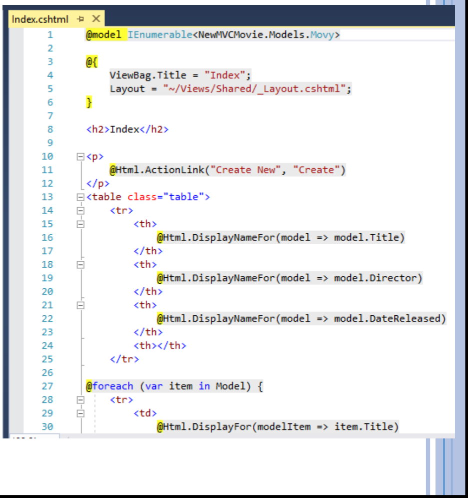
☑ Use a layout page:

~/Views/Shared/_Layout.cshtml

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

Closer look at generated List view

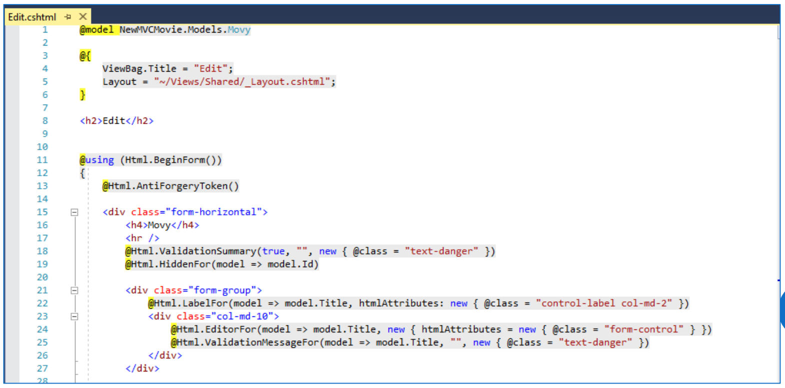
- First line
  - @model Ienumerable <NewMVCMovie.Models.Movy>
  - Allows you to access the list of movies that the controller passed to the view



```
@model IEnumerable<NewMVCMovie.Models.Movy>

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Title)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Director)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.DateReleased)
        </th>
        <th></th>
    </tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Title)
```

Closer look at generated Edit view

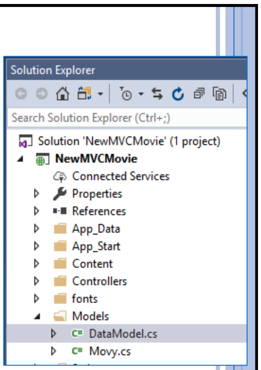
- First line
  - @model NewMVCMovie.Models.Movy
  - Specifies the type of object the view expects
  - Allows you to access the object that the controller passed to the view

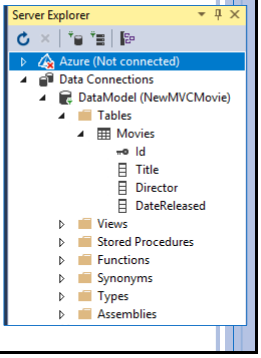


```
@model NewMVCMovie.Models.Movy

@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Edit</h2>


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

    <div class="form-horizontal">
        <h4>Movy</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)

        <div class="form-group">
            @Html.LabelFor(model => model.Title, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Title, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Title, "", new { @class = "text-danger" })
            </div>
        </div>
```

Working with SQL LocalDB

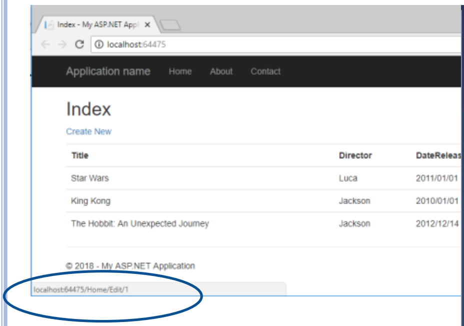
- Must be in App_Data folder
- Can be created from within VS
  - Server Explorer, Add item, SQL Server Database
  - Add table with fields
  - Add records
  - Use wizard to create model (classes)
- Class can also be created first
  - Database created based on created class
  - Might have to install Entity Framework (Tools, NuGet Package Manager, Manage NuGet Packages for Solution)
- Use Server Explorer
  - Open tables
  - Open table definition





Examining the Index view

- Hover on edit link (of first item)
  - http://localhost:64475/Home/Edit/1



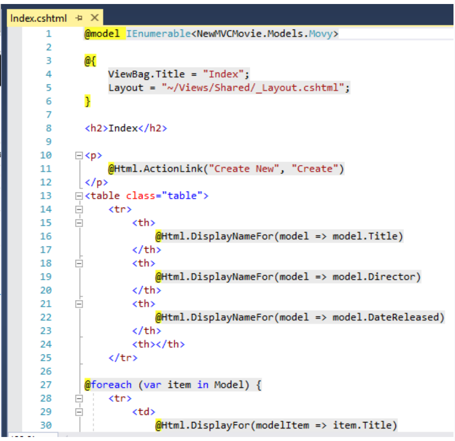
Index

Create New

Title	Director	DateReleas
Star Wars	Luca	2011/01/01
King Kong	Jackson	2010/01/01
The Hobbit: An Unexpected Journey	Jackson	2012/12/14

© 2018 - My ASP.NET Application

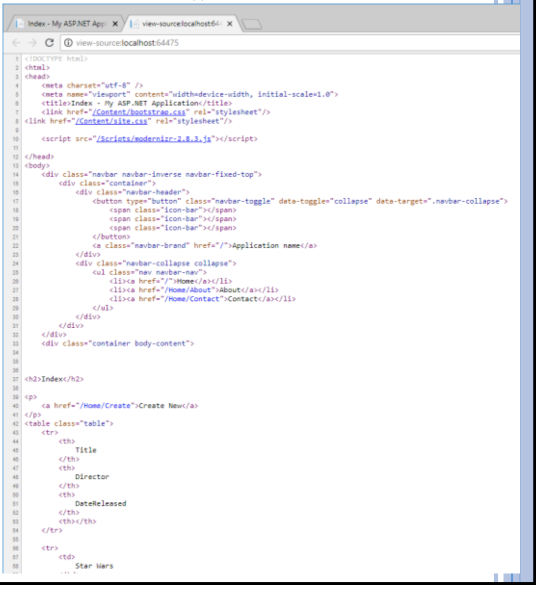
localhost:64475/Home/Edit/1



```
@model IEnumerable<NewMVCMovie.Models.Movy>

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Title)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Director)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.DateReleased)
        </th>
        <th></th>
    </tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Title)
```

EXAMINING THE RENDERED PAGE

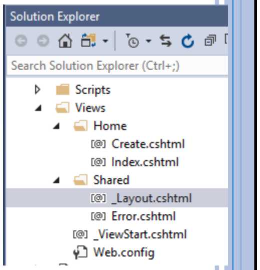
- Run application
- In the browser, view the source
  - View the page as rendered by the browser

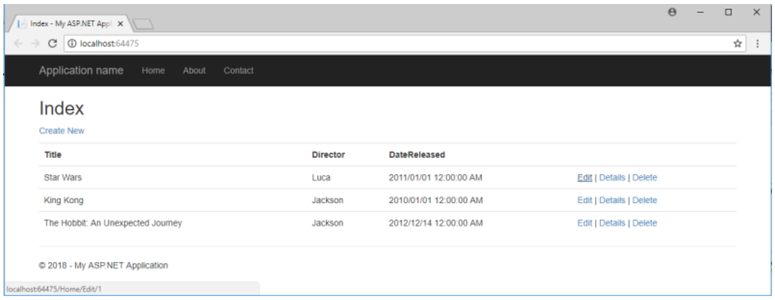


```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Index - My ASP.NET Application</title>
    <link href="/Content/bootstrap.css" rel="stylesheet"/>
<link href="/Content/site.css" rel="stylesheet"/>

    <script src="/Scripts/modernizr-2.8.3.js"></script>

</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="/">Application name</a>
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li><a href="/">Home</a></li>
                    <li><a href="/Home/About">About</a></li>
                    <li><a href="/Home/Contact">Contact</a></li>
                </ul>
            </div>
        </div>
    </div>
    <div class="container body-content">



<h2>Index</h2>

<p>
    <a href="/Home/Create">Create New</a>
</p>
<table class="table">
    <tr>
        <th>
            Title
        </th>
        <th>
            Director
        </th>
        <th>
            DateReleased
        </th>
        <th></th>
    </tr>

    <tr>
        <td>
            Star Wars
```

SPECIFY STYLE AND LAYOUT

- There is no master page
  - Layout file functions as the blueprint for multiple pages
- In the Shared folder: _Layout.cshtml file





Index

Create New

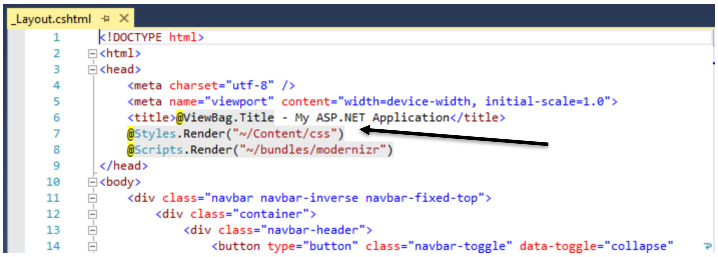
Title	Director	DateReleased	
Star Wars	Luca	2011/01/01 12:00:00 AM	Edit | Details | Delete
King Kong	Jackson	2010/01/01 12:00:00 AM	Edit | Details | Delete
The Hobbit: An Unexpected Journey	Jackson	2012/12/14 12:00:00 AM	Edit | Details | Delete

© 2018 - My ASP.NET Application

localhost:64475/Home/Edit/1

Specify style and layout

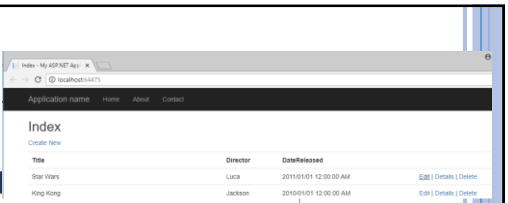
- In the Shared folder: _Layout.cshtml file
  - Css still in head element



_Layout.cshtml

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse"
```

<Body>



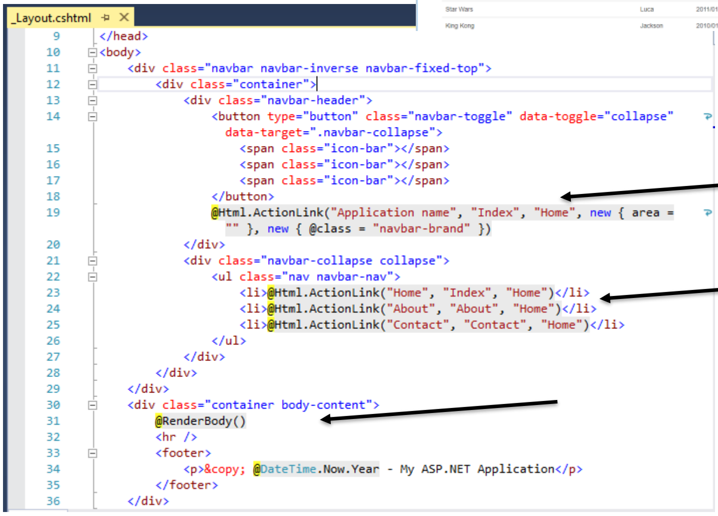
Index - My ASP.NET App

localhost:64475

Index

Create New

Title	Director	DateReleased	
Star Wars	Luca	2011/01/01 12:00:00 AM	Edit | Details | Delete
King Kong	Jackson	2010/01/01 12:00:00 AM	Edit | Details | Delete

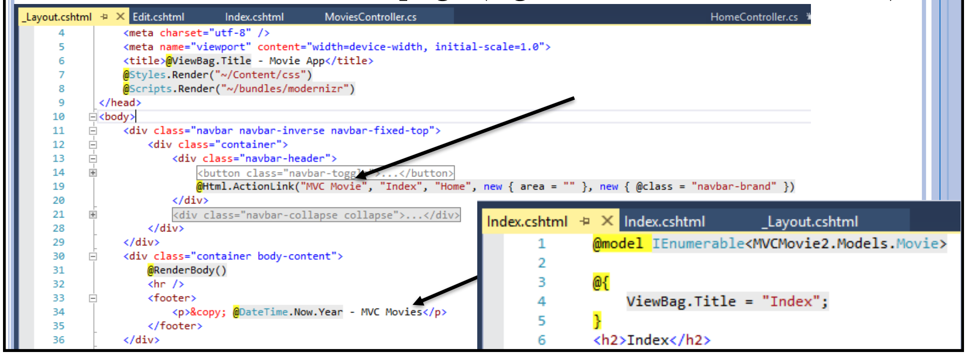


_Layout.cshtml

```
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse"
                  data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                @Html.ActionLink("Application name", "Index", "Home", new { area =
                  "" }, new { @class = "navbar-brand" })
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li>@Html.ActionLink("Home", "Index", "Home")</li>
                    <li>@Html.ActionLink("About", "About", "Home")</li>
                    <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
                </ul>
            </div>
        </div>
    </div>
    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
        </footer>
    </div>
```

STYLE AND LAYOUT

- Some elements (text or logos) in _Layout.cshtml page should be adapted (if you used a template)
  - E.g. My Web Application should be MVC Movie or Movie App
- Some values in view page (e.g. title – see index.cshtml)

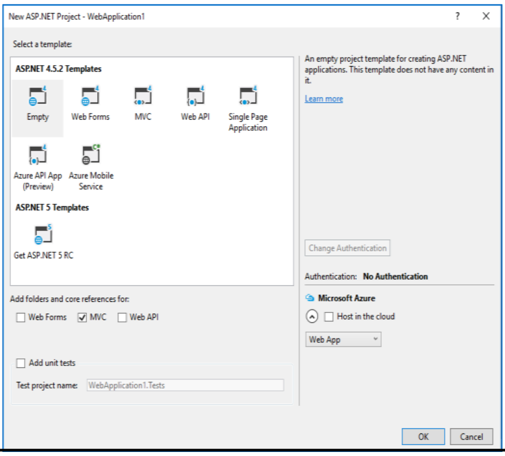


```
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>@ViewBag.Title - Movie App</title>
        @Styles.Render("~/Content/css")
        @Scripts.Render("~/bundles/modernizr")
    </head>
    <body>
        <div class="navbar navbar-inverse navbar-fixed-top">
            <div class="container">
                <div class="navbar-header">
                    <button class="navbar-toggle">...</button>
                    @Html.ActionLink("MVC Movie", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
                </div>
                <div class="navbar-collapse collapse">...</div>
            </div>
        </div>
        <div class="container body-content">
            @RenderBody()
            <hr />
            <footer>
                <p>&copy; @DateTime.Now.Year - MVC Movies</p>
            </footer>
        </div>
```

```
@model IEnumerable<MVCMovie2.Models.Movie>

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

MODEL-VIEW-CONTROLLER (MVC)

- File, New, **Project**; ASP.NET Web Application
  - Do you require a empty start-up application?


New ASP.NET Project - WebApplication1

Select a template:

ASP.NET 4.5.2 Templates

ASP.NET 5 Templates

An empty project template for creating ASP.NET applications. This template does not have any content in it.

Learn more

Authentication: **No Authentication**

Add folders and core references for:

□ Web Forms ☑ MVC □ Web API

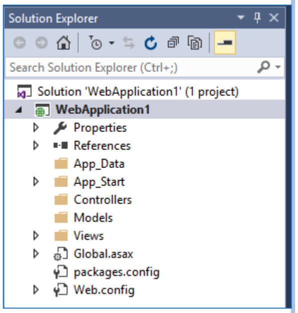
Microsoft Azure

□ Host in the cloud

Web App

□ Add unit tests

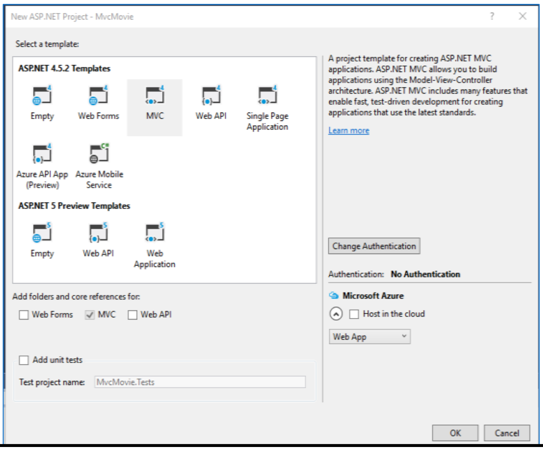
Test project name: WebApplication1.Tests



Solution 'WebApplication1' (1 project)
- **WebApplication1**
  - Properties
  - References
  - App_Data
  - App_Start
  - Controllers
  - Models
  - Views
  - Global.asax
  - packages.config
  - Web.config

Model-View-Controller (MVC)

- File, New, **Project**; ASP.NET Web Application
  - Would you like a working project to edit?



New ASP.NET Project - MvcMovie

Select a template:

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Authentication: **No Authentication**

Microsoft Azure

□ Host in the cloud

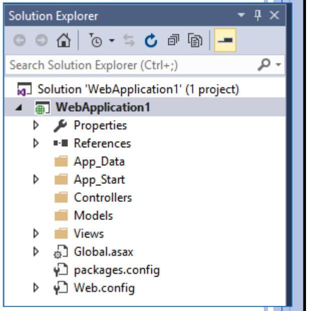
Web App

Add folders and core references for:

□ Web Forms ☑ MVC □ Web API

□ Add unit tests

Test project name: MvcMovie.Tests



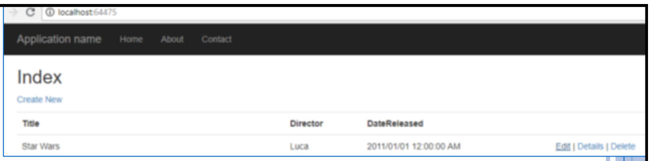
Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'WebApplication1' (1 project)
  WebApplication1
    Properties
    References
    App_Data
    App_Start
    Controllers
    Models
    Views
    Global.asax
    packages.config
    Web.config
```

Using a Working project as start-up?

- Remember to remove the controllers not required
- Check all the "template" headings and data, and replace with appropriate data



Index

Create New

Title	Director	DateReleased	
Star Wars	Luca	2011/01/01 12:00:00 AM	Edit | Details | Delete

PRACTICAL 07

- Create a new MVC web application with no authentication:
  - Keep track of your favourite books/music collection
- Must have a database – single table sufficient
  - Table must have at least 4 fields
- Controller to handle requests and link to model
- Functional Index, Create and Edit Methods and Views
- Edit main menu option text – make them more relevant
  - Only the Home link need to be functional at this stage
- Change the black menu background to a different colour

RESOURCES

- Why use MVC: https://www.codeproject.com/Articles/821275/Webforms-vs-MVC-and-Why-MVC-is-better
- Bundling in MVC: https://www.c-sharpcorner.com/UploadFile/17e8f6/bundling-in-mvc4/
- Routing Basics in ASP.NET MVC: http://johnatten.com/2013/07/21/routing-basics-in-asp-net-mvc/
- Customizing routes in MVC https://www.codeproject.com/Articles/641783/Customizing-Routes-in-ASP-NET-MVC
- **MVC Application Tutorial**
- **Intro to ASP.NET MVC 4** with Visual Studio