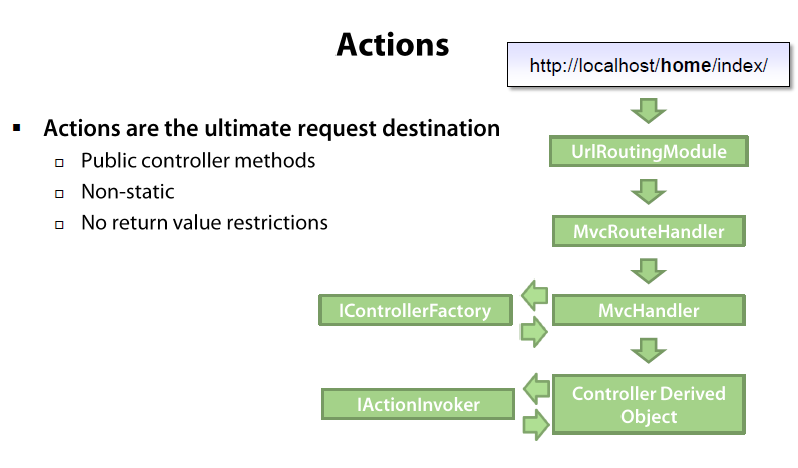
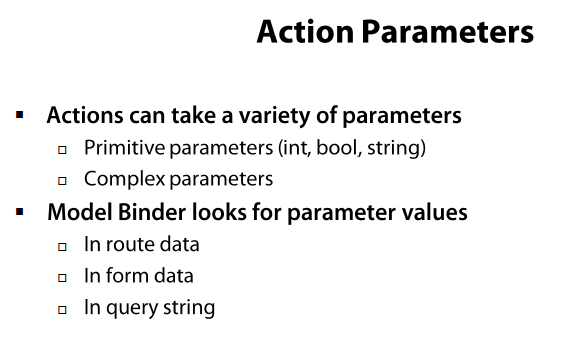


****





**ActionResult**

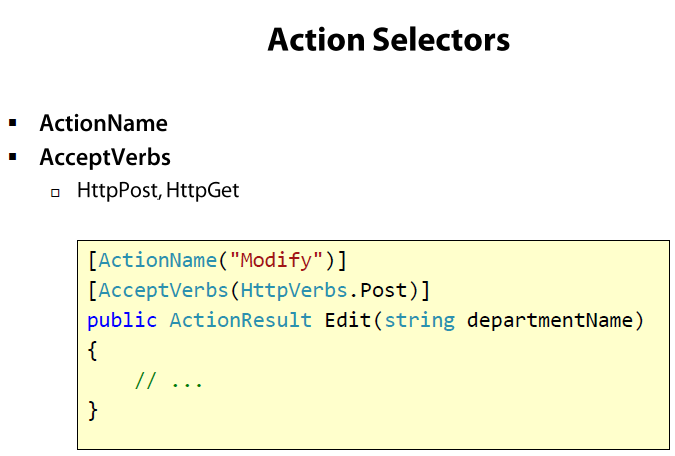
ActionResult is an abstract class that can have several subtypes.

**ActionResult Subtypes**

* **ViewResult** - Renders a specified view to the response stream
* **PartialViewResult** - Renders a specified partial view to the response stream
* **EmptyResult** - An empty response is returned
* **RedirectResult** - Performs an HTTP redirection to a specified URL
* **RedirectToRouteResult** - Performs an HTTP redirection to a URL that is determined by the routing engine, based on given route data
* **JsonResult** - Serializes a given ViewData object to JSON format
* **JavaScriptResult** - Returns a piece of JavaScript code that can be executed on the client
* **ContentResult** - Writes content to the response stream without requiring a view
* **FileContentResult** - Returns a file to the client
* **FileStreamResult** - Returns a file to the client, which is provided by a Stream
* **FilePathResult** - Returns a file to the client

# [Custom ASP.NET MVC ActionResults](http://www.codeproject.com/Articles/533932/Custom-ASP-NET-MVC-ActionResults)

# [Actions in ASP.NET MVC 3](http://www.codeproject.com/Articles/195434/Actions-in-ASP-NET-MVC-3)



## ASP.NET Razor

[**Razor Intro**](http://www.w3schools.com/aspnet/razor_intro.asp)

[Razor Syntax](http://www.w3schools.com/aspnet/razor_syntax.asp)

[Razor C# Variables](http://www.w3schools.com/aspnet/razor_cs_variables.asp)

[Razor C# Loops](http://www.w3schools.com/aspnet/razor_cs_loops.asp)

[Razor C# Logic](http://www.w3schools.com/aspnet/razor_cs_logic.asp)

[MVC HTML Helpers](http://www.w3schools.com/aspnet/mvc_htmlhelpers.asp)

### [Introduction to ASP.NET Web Programming Using the Razor Syntax (C#)](http://www.asp.net/web-pages/tutorials/basics/2-introduction-to-asp-net-web-programming-using-the-razor-syntax)

## What is Razor?

Razor is a markup syntax that lets you embed server-based code (C#) into web pages.

Server-based code can create dynamic web content on the fly, while a web page is written to the browser. When a web page is called, the server executes the server-based code inside the page before it returns the page to the browser. By running on the server, the code can perform complex tasks, like accessing databases.

## Razor Syntax

<ul>

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

<li>@i</li>

}

</ul>

Main Razor Syntax Rules for C#

* Razor code blocks are enclosed in @{ ... }
* Inline expressions (variables and functions) start with @
* Code statements end with semicolon
* Variables are declared with the var keyword
* Strings are enclosed with quotation marks
* C# code is case sensitive
* C# files have the extension .cshtml

## C# Example

<!-- 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>

# ASP.NET Razor - C# Variables

## Examples

@{

// Assigning a string to a variable.

var greeting = "Welcome!";

// Assigning a number to a variable.

var theCount = 3;

// Assigning an expression to a variable.

var monthlyTotal = theCount + 5;

// Assigning a date value to a variable.

var today = DateTime.Today;

// Assigning the current page's URL to a variable.

var myPath = this.Request.Url;

// Declaring variables using explicit data types.

string name = "Joe";

int count = 5;

DateTime tomorrow = DateTime.Now.AddDays(1);

}

## Converting Data Types

Converting from one data type to another is sometimes useful.  
  
The most common example is to convert string input to another type, such as an integer or a date.

As a rule, user input comes as strings, even if the user entered a number. Therefore, numeric input values must be converted to numbers before they can be used in calculations.

Below is a list of common conversion methods:

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Example** |
| AsInt() IsInt() | Converts a string to an integer. | if (myString.IsInt())   {myInt=myString.AsInt();} |
| AsFloat() IsFloat() | Converts a string to a floating-point number. | if (myString.IsFloat())   {myFloat=myString.AsFloat();} |
| AsDecimal() IsDecimal() | Converts a string to a decimal number. | if (myString.IsDecimal())   {myDec=myString.AsDecimal();} |
| AsDateTime() IsDateTime() | Converts a string to an ASP.NET DateTime type. | myString="10/10/2012"; myDate=myString.AsDateTime(); |
| AsBool() IsBool() | Converts a string to a Boolean. | myString="True"; myBool=myString.AsBool(); |
| ToString() | Converts any data type to a string. | myInt=1234; myString=myInt.ToString(); |

# ASP.NET MVC - HTML Helpers

HTML Helpers are used to modify HTML output

## HTML Links

The easiest way to render an HTML link in is to use the HTML.ActionLink() helper.

With MVC, the Html.ActionLink() does not link to a view. It creates a link to a controller action.

Razor Syntax:

@Html.ActionLink("About this Website", "About")

Razor Syntax C#:

@Html.ActionLink("Edit Record", "Edit", new {Id=3})

The Html.ActionLink() helper above, outputs the following HTML:

<a href="/Home/Edit/3">Edit Record</a>

HTML Form Elements

There following HTML helpers can be used to render (modify and output) HTML form elements:

* BeginForm()
* EndForm()
* TextArea()
* TextBox()
* CheckBox()
* RadioButton()
* ListBox()
* DropDownList()
* Hidden()
* Password()

|  |  |
| --- | --- |
| **Helper** | **HTML Element** |
| Html.CheckBox | <input type="checkbox" /> |
| Html.DropDownList | <select></select> |
| Html.Hidden | <input type="hidden" /> |
| Html.Label | <label for="" /> |
| Html.ListBox | <select></select> or <select multiple></select> |
| Html.Password | <input type="password" /> |
| Html.Radio | <input type="radio" /> |
| Html.TextArea | <textarea></textarea> |
| Html.TextBox | <input type="text" /> |

|  |  |  |
| --- | --- | --- |
| **Method** | **Action** | **Output** |
| **ActionLink(s:text, s:action, o:attributes)** | Writes an anchor tag to a link for a specific action. | <a href="action">text</a> |
| **AntiForgeryToken(s:salt, s:domain, s:path)** | Generates a hidden form field (anti-forgery token) that is validated when the form is submitted. |  |
| **AttributeEncode(s: input)** | HTML-encodes the string (as an attribute). |  |
| **BeginForm(s:action, s:controller, o:values)** | Writes an opening <form> tag to the response. | <form action="/controller/action/"> |
| **BeginRouteForm(s:routeName)** | Writes an opening <form> tag for the route. | <form action="route"> |
| **CheckBox(s:name, b:checked)** | Returns a check box input element. | <input type="checkbox" name="name" id="name" checked="checked" /> |
| **CheckBoxFor(e:expression)** | Returns a check box input element for the model. | <input type="checkbox" name="name" id="name" checked="checked" /> |
| **DropDownList(s:name, list:selectlistitems)** | Returns a single-selection select element. | <select name="name" id="name"></select> |
| **DropDownListFor(e:expression, list:selectlistitems)** | Returns a single-selection select element for the model. | <select name="name" id="name"></select> |
| **Encode(s:input)** | HTML-encodes the string. |  |
| **EndForm()** | Renders the closing </form> tag to the response. | </form> |
| **Hidden(s:name, o:value)** | Returns a hidden input element. | <input type="hidden" value="value" name="name" /> |
| **HiddenFor(e:expression)** | Returns a hidden input element for the model. | <input type="hidden" value="value" name="name" /> |
| **ListBox(s:name, list:selectlistitems)** | Returns a multi-select select element. | <select multiple="multiple" name="name" id="name"></select> |
| **ListBoxFor(e:expression, list:selectlistitems)** | Returns a multi-select select element for the model. | <select multiple="multiple" name="name" id="name"></select> |
| **Password(s:name, o:value)** | Returns a password input element. | <input type="password" value="value" name="name" /> |
| **PasswordFor(e:expression)** | Returns a password input element for the model. | <input type="password" value="value" name="name" /> |
| **RadioButton(s:name, o:value, b:checked)** | Returns a radio button input element. | <input type="radio" value="value" name="name" checked="checked" /> |
| **RadioButtonFor(e:expression, o:value)** | Returns a radio button input element for the model. | <input type="radio" value="value" name="name" checked="checked" /> |
| **Partial(s:name, o:model)** | Renders a partial view (.cshtml). |  |
| **RouteLink(s:text, s:routeName)** | Returns an anchor element (a element) that contains the virtual path of the specified action. | <a href="action">text</a> |
| **TextArea(s:name, s:value)** | Returns the specified textarea element. | <textarea name="name">value</textarea> |
| **TextAreaFor(e:expression)** | Returns the specified textarea element for the model. | <textarea name="name">value</textarea> |
| **TextBox(s:name, o:value)** | Returns a text input element. | <input type="text" name="name" value="value" /> |
| **TextBoxFor(e:expression)** | Returns a text input element for the model. | <input type="text" name="name" value="value" /> |
| **TextBoxFor(e:expression)** | Returns a text input element for the model. | <input type="text" name="name" value="value" /> |

**UrlHelper**

|  |  |  |
| --- | --- | --- |
| **Method** | **Action** | **Output** |
| **Action(s:action, s:controller)** | Generates a fully qualified URL to an action method. |  |
| **Content(s:path)** | Converts a virtual (relative) path to an application absolute path. |  |
| **Encode(s:url)** | Encodes special characters in a URL string into character-entity equivalents. |  |
| **RouteUrl(s:route)** | Generates a fully qualified URL for the specified route name. |  |

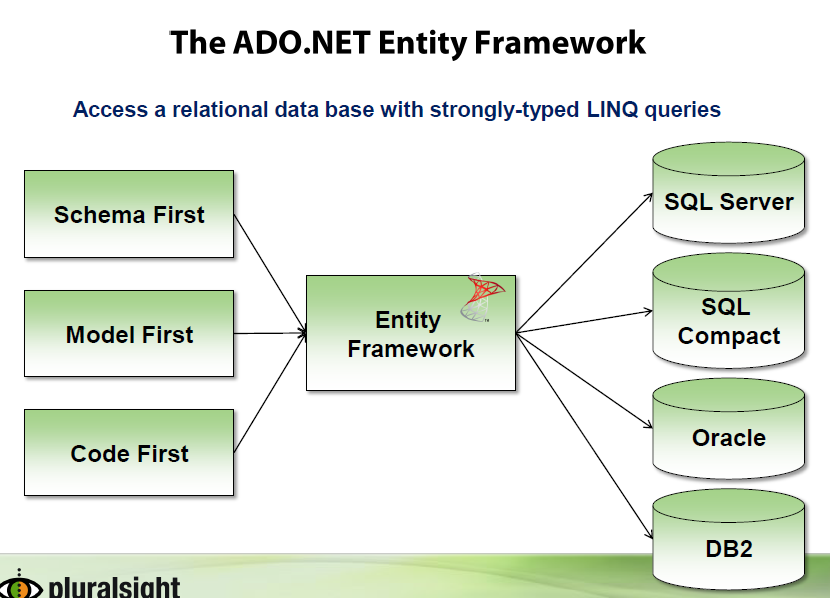
### [Creating Custom HTML Helpers - ASP.NET MVC 4](http://passionatetechie.blogspot.co.uk/2013/03/creating-custom-html-helpers-aspnet-mvc.html)

[ASP.NET MVC 3 : Creating Custom HTML Helpers](http://www.aspnetwiki.com/page:creating-custom-html-helpers)

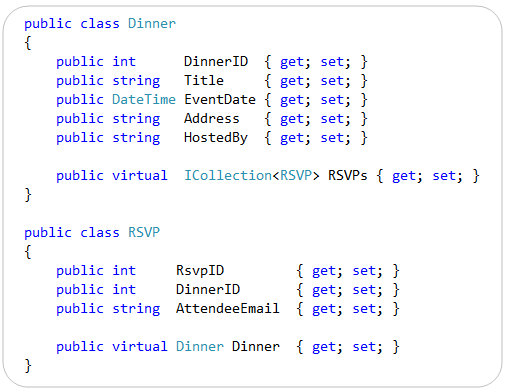
### [EF Feature CTP4 Walkthrough: Code First](http://blogs.msdn.com/b/adonet/archive/2010/07/14/ctp4codefirstwalkthrough.aspx)

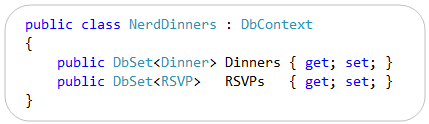
For Code First to a New Database see <http://msdn.com/data/jj193542>

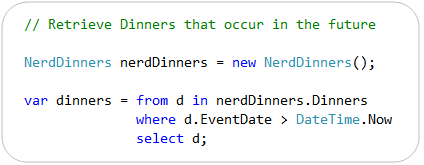
For Code First to an Existing Database see <http://msdn.com/data/jj200620>

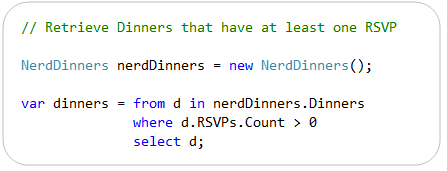


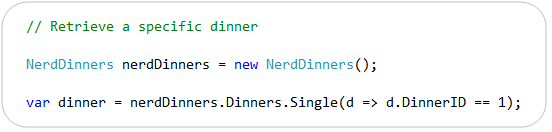
## [Code-First Development with Entity Framework 4](http://weblogs.asp.net/scottgu/archive/2010/07/16/code-first-development-with-entity-framework-4.aspx)

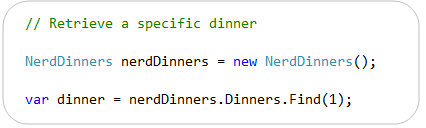


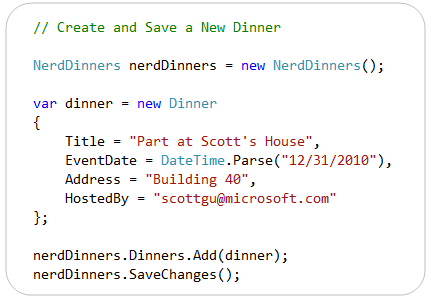


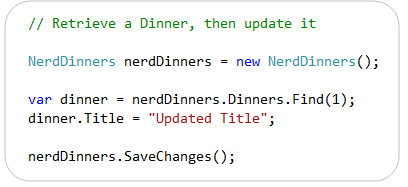




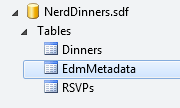


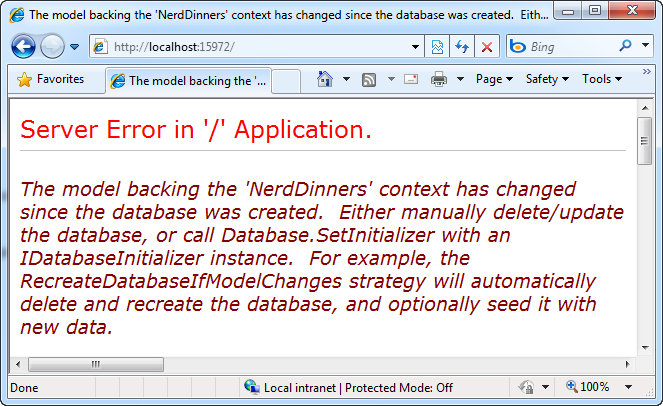






EF automatically creates a database for you, it by default adds an “EdmMetadata” table to the database that tracks the shape of the model objects that were used to automatically create the database schema for you:

[](http://weblogs.asp.net/blogs/scottgu/image_6AF29321.png)



The error message above occurs when EF detects that you’ve made a change to a model object and it is now out of sync with the database it automatically created for you.

Re-synchronizing our Model Classes with the Database

There are a couple of ways we can “re-sync” our model objects and our database:

* We can manually update our database schema to match our models
* We can manually delete our database file, re-run the application, and have EF automatically re-create the database
* We can enable a feature of EF code-first that automatically updates our database for us whenever we change our models

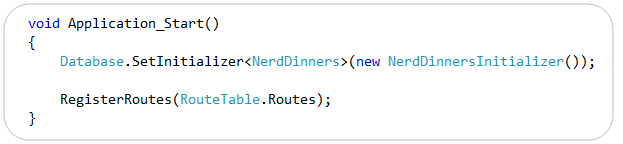
The RecreateDatabaseIfModelChanges Feature



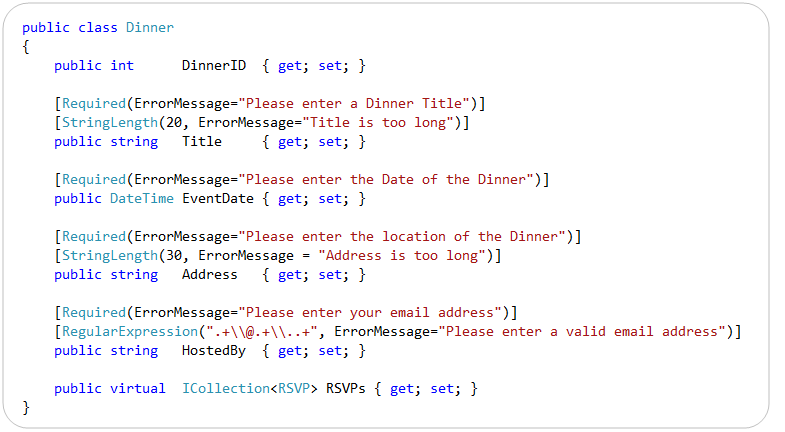
Seeding Initial Data in Automatically Created Databases



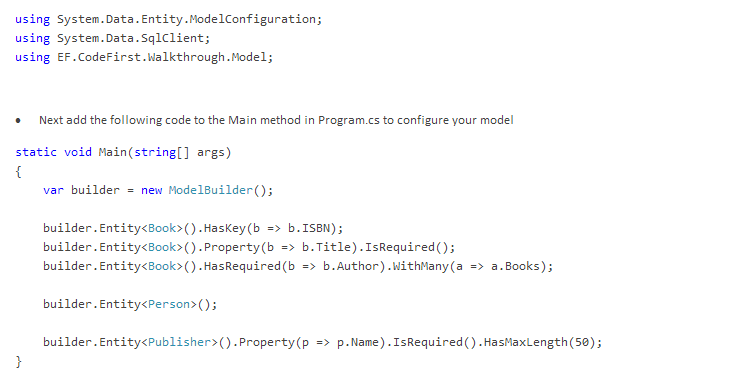
We can then update the Database.Initializer() call we added to our Global.asax to use this “NerdDinnersInitializer” class at startup:

[](http://weblogs.asp.net/blogs/scottgu/image_04B6A384.png)

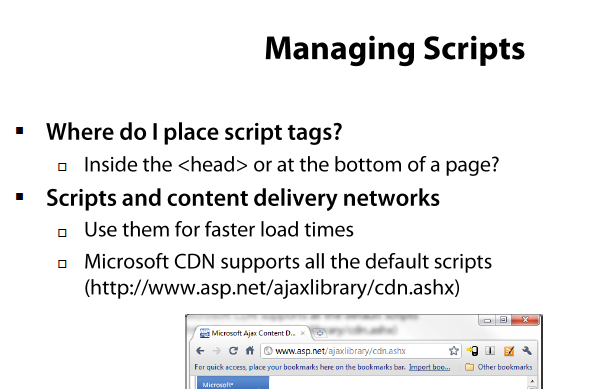
Adding Validation using DataAnnotations



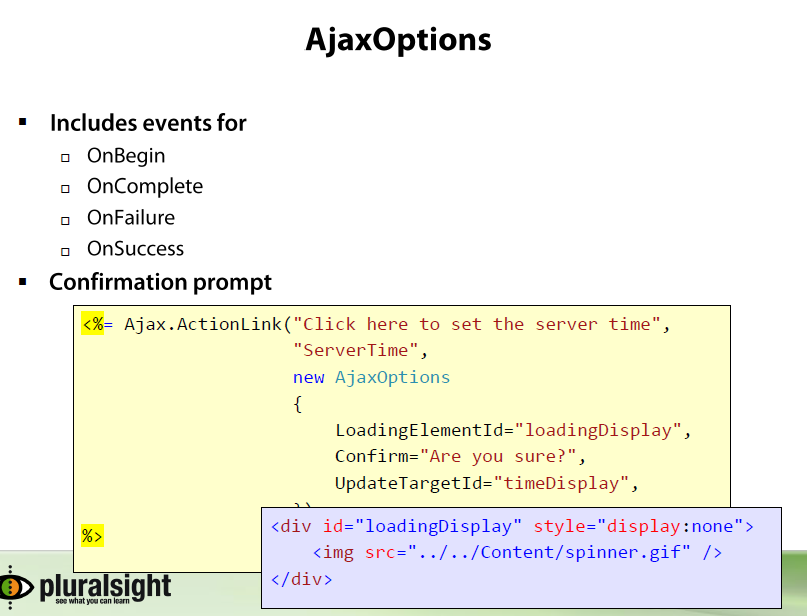
# Model Builder











This will render a link with displayed text “Click here.” When the user clicks on the link the “**ServerTime” action** will

be invoked via Ajax. The response from this action (probably some HTML fragment) will be placed in an element

with id “TimeDisplay.” The available parameters you can pass to the AjaxOptions class to customize the behavior of the link are:

**HttpMethod** can be “GET” or “POST”. The default is ”GET”

**UpdateTargetId** The element that will receive the content

**InsertionMode** Can be InsertBefore, InsertAfter, or Replace

**OnBegin** Javascript function to be called before invoking the action

**OnComplete** Javascript function to be called after the response comes back

**OnFailure** Javascript function to be called in the event of an error

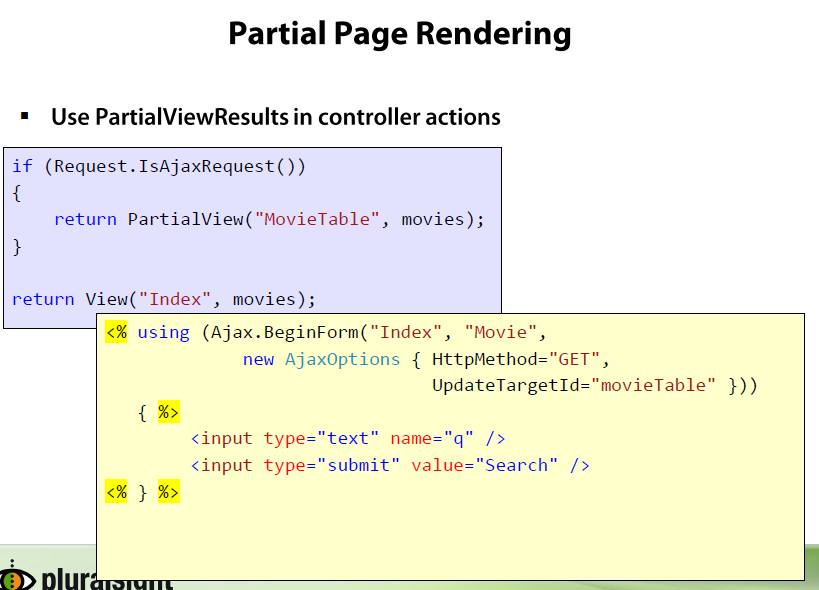
**OnSuccess** Javascript function to be called if no errors occur

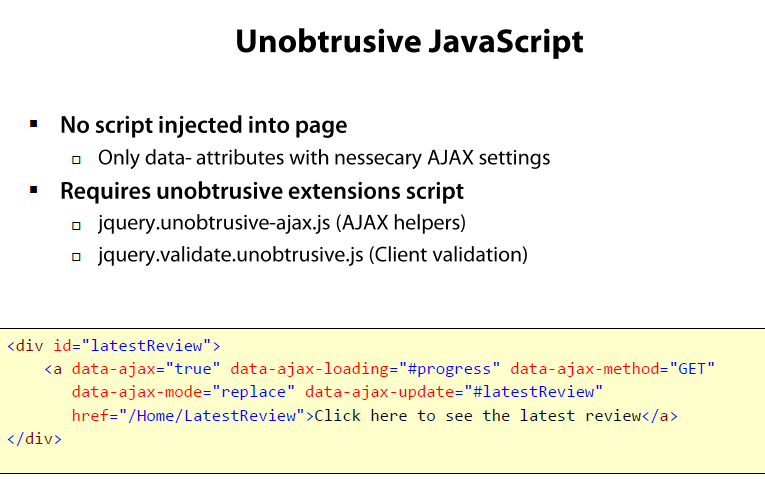
**Confirm** Confirmation message to provide an OK/Cancel dialog before proceeding

**Url** Url to use if the anchor tag has a different destination than the Ajax request

**LoadingElementId** An element that displays Ajax progress. The element should be marked as visibility:hidden initially.

[**Working with Ajax Helper in ASP.NET MVC**](http://www.codeguru.com/csharp/.net/working-with-ajax-helper-in-asp.net-mvc.htm)





## Enabling Client-Side Validation

To enable client-side validation in ASP.NET MVC 3, you must set two flags and you must include three JavaScript files.

Open the application's Web.config file. Verify that ClientValidationEnabled andUnobtrusiveJavaScriptEnabled are set to true in the application settings. The following fragment from the rootWeb.config file shows the correct settings:

<appSettings>

<add key="ClientValidationEnabled" value="true"/>

<add key="UnobtrusiveJavaScriptEnabled" value="true"/>

</appSettings>

Setting UnobtrusiveJavaScriptEnabled to true enables unobtrusive Ajax and unobtrusive client validation. When you use unobtrusive validation, the validation rules are turned into HTML5 attributes. HTML5 attribute names can consist of only lowercase letters, numbers, and dashes.

Setting ClientValidationEnabled to true enables client-side validation. By setting these keys in the applicationWeb.config file, you enable client validation and unobtrusive JavaScript for the entire application. You can also enable or disable these settings in individual views or in controller methods using the following code:

HtmlHelper.ClientValidationEnabled = true;

HtmlHelper.UnobtrusiveJavaScriptEnabled = true;

You also need to include several JavaScript files in the rendered view. An easy way to include the JavaScript in all views is to add them to the *Views\Shared\\_Layout.cshtml* file. Replace the <head> element of the *\_Layout.cshtml* file with the following code:

<head>

<title>@View.Title</title>

<link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />

<script src="http://ajax.microsoft.com/ajax/jQuery/jquery-1.4.2.min.js"></script>

<script src="http://ajax.microsoft.com/ajax/jquery.validate/1.7/jquery.validate.min.js"></script>

<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")" type="text/javascript"></script>

</head>

Run the application and click an edit link. View the page's source in the browser. The browser source shows many attributes of the form data-val (for data validation). When client validation and unobtrusive JavaScript is enabled, input fields with a client-validation rule contain the data-val="true" attribute to trigger unobtrusive client validation. For example, the City field in the model was decorated with the [Required](http://msdn.microsoft.com/en-us/library/system.componentmodel.dataannotations.requiredattribute.aspx) attribute, which results in the HTML shown in the following example:

<div class="editor-field">

<input data-val="true" data-val-required="The City field is required." id="City" name="City" type="text" value="Seattle" />

<span class="field-validation-valid" data-valmsg-for="City" data-valmsg-replace="true"></span>

</div>

<input data-val="true"

data-val-length="The field First Name must be a string with a minimum length of 3 and a maximum length of 8."

data-val-length-max="8"

data-val-length-min="3"

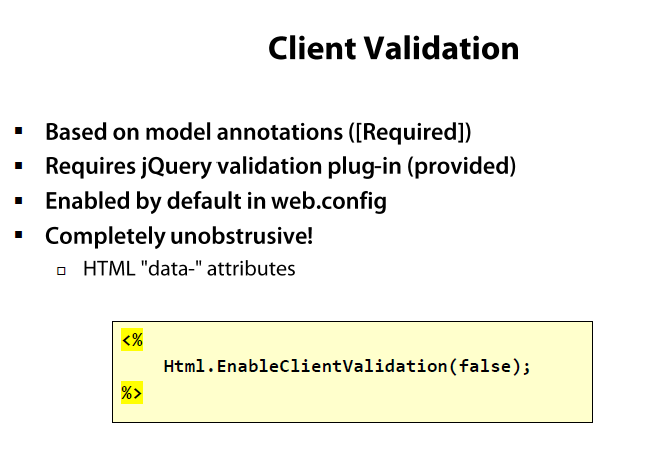
data-val-required="The First Name field is required."

id="FirstName"

name="FirstName"

type="text"

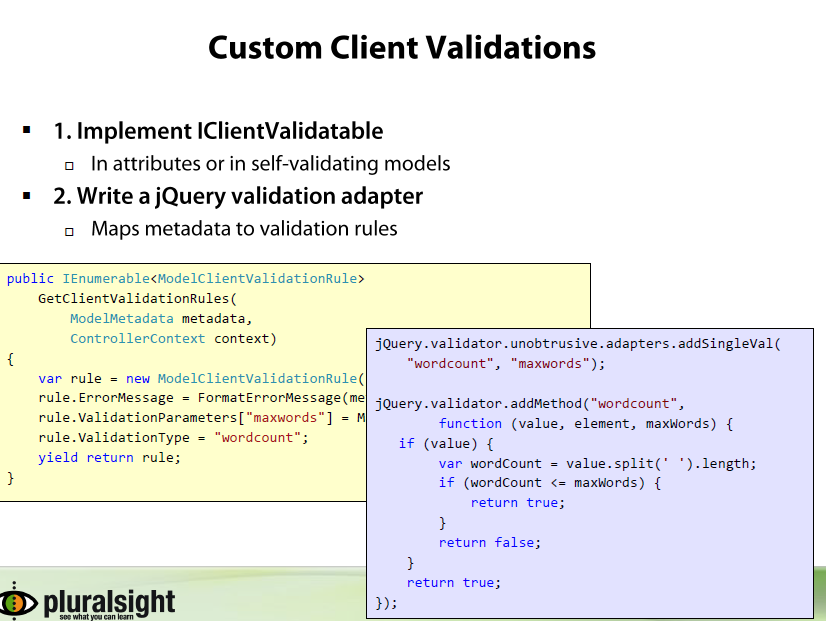
value="Ben" />



[Creating a MVC 3 Application with Razor and Unobtrusive JavaScript](http://www.asp.net/mvc/tutorials/older-versions/javascript/creating-a-mvc-3-application-with-razor-and-unobtrusive-javascript)

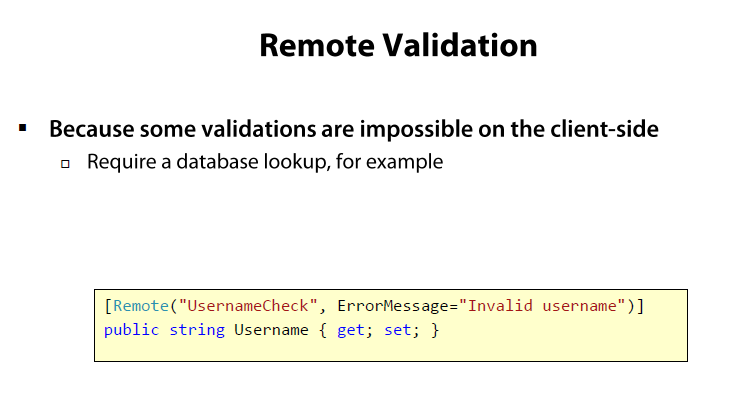
### [Unobtrusive Client Validation in ASP.NET MVC 3](http://bradwilson.typepad.com/blog/2010/10/mvc3-unobtrusive-validation.html)

# [Validating Model Data and Unobtrusive Client side Validation in ASP.NET MVC](http://www.codeproject.com/Articles/577937/A-Beginners-Tutorial-on-Validating-Model-Data-and)



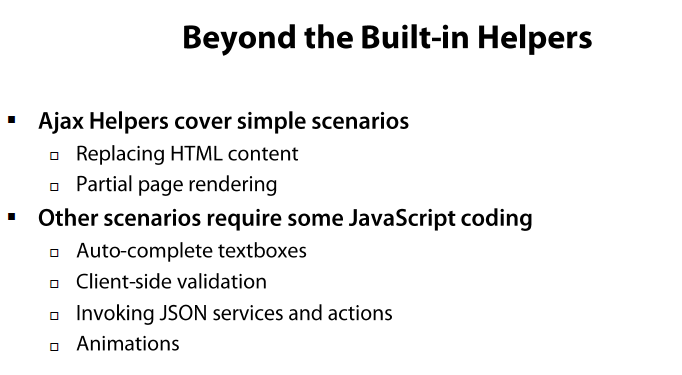
# [Custom Client Side Validation in ASP.NET MVC3](http://www.codeproject.com/Articles/275056/Custom-Client-Side-Validation-in-ASP-NET-MVC3)

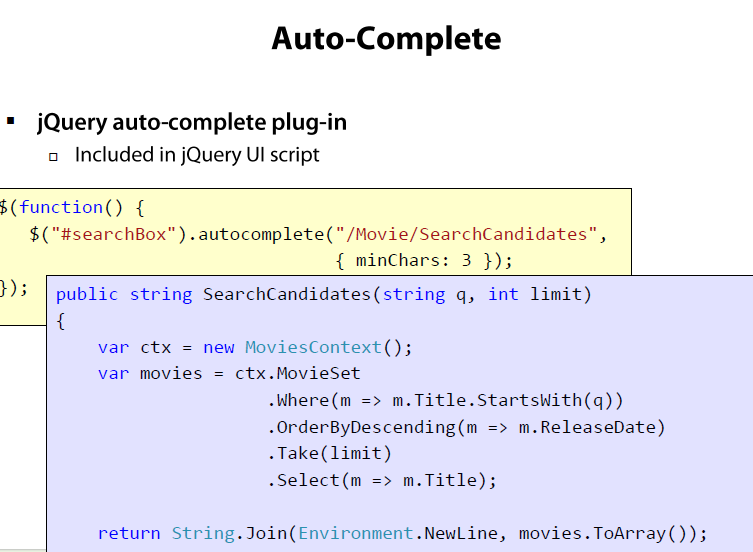
# [Building Client (JavaScript) Custom Validation in ASP.NET MVC 4 using jQuery](http://www.codeproject.com/Articles/613330/Building-Client-JavaScript-Custom-Validation-in-AS)



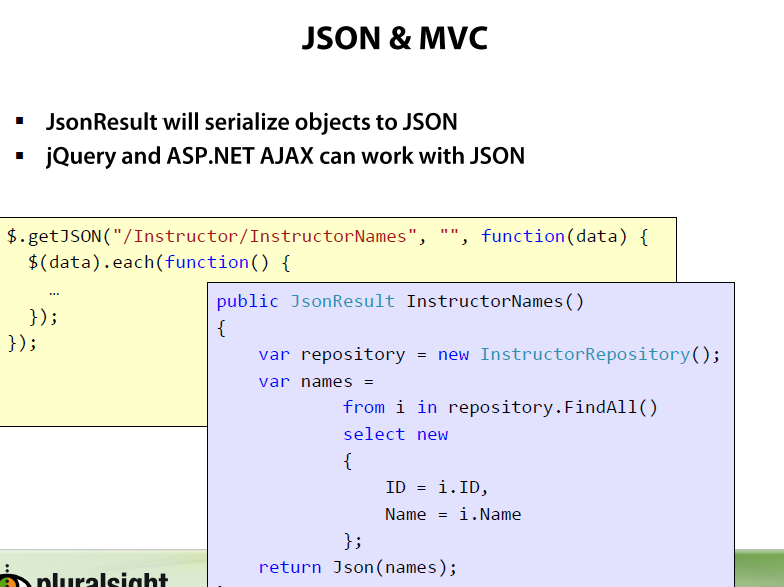
# [How to: Implement Remote Validation from a Client in MVC](http://msdn.microsoft.com/en-us/library/ff398048(v=vs.100).aspx)

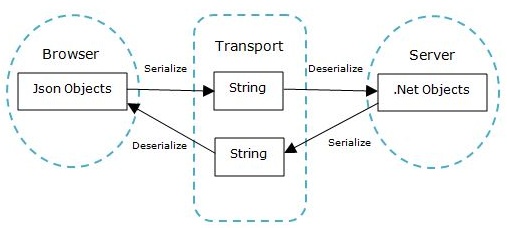
# [How to: Implement Remote Validation in ASP.NET MVC](http://msdn.microsoft.com/en-us/library/gg508808(v=vs.98).aspx)



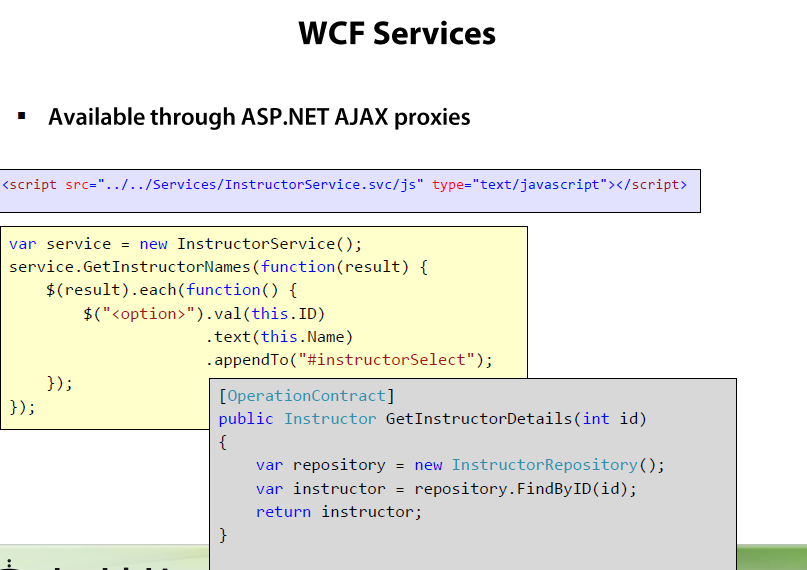








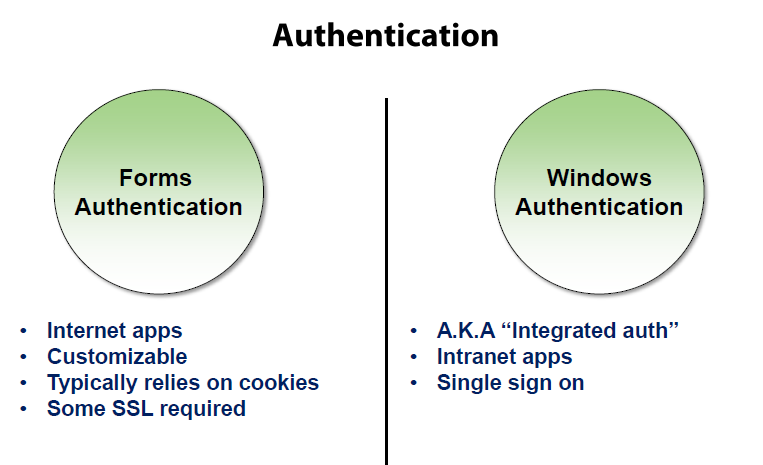
# [Working on JSON objects in jQuery and MVC](http://www.codeproject.com/Articles/124541/Working-on-JSON-objects-in-jQuery-and-MVC)

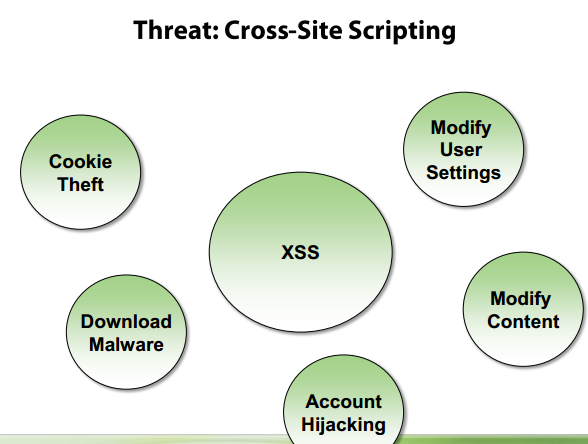


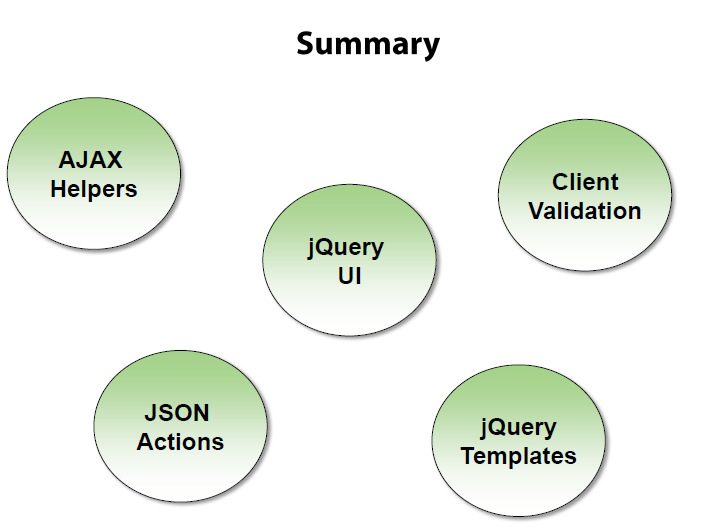
# [MVC3 With Ajax Enabled WCF](http://code.msdn.microsoft.com/MVC3-With-Ajax-Enabled-WCF-11957be0)

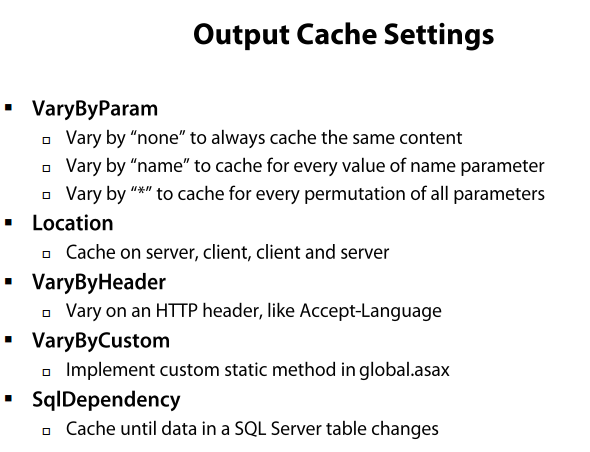
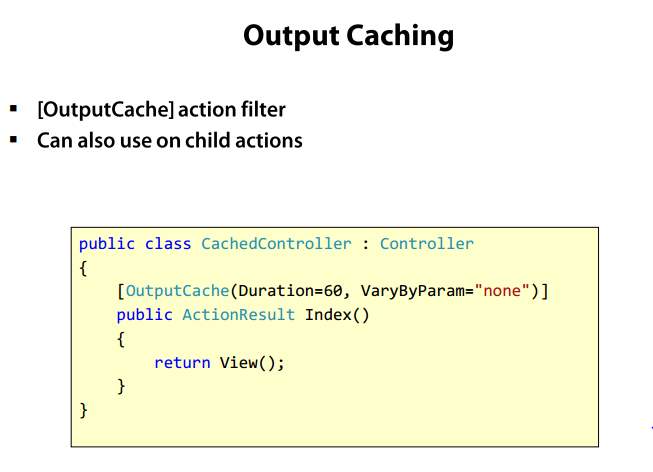
### [Steps to Call WCF Service using jQuery](http://pranayamr.blogspot.co.uk/2010/12/steps-to-call-wcf-service-using-jquery.html)





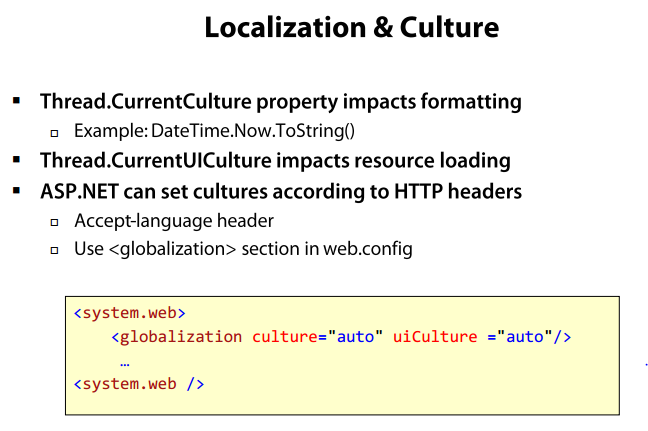


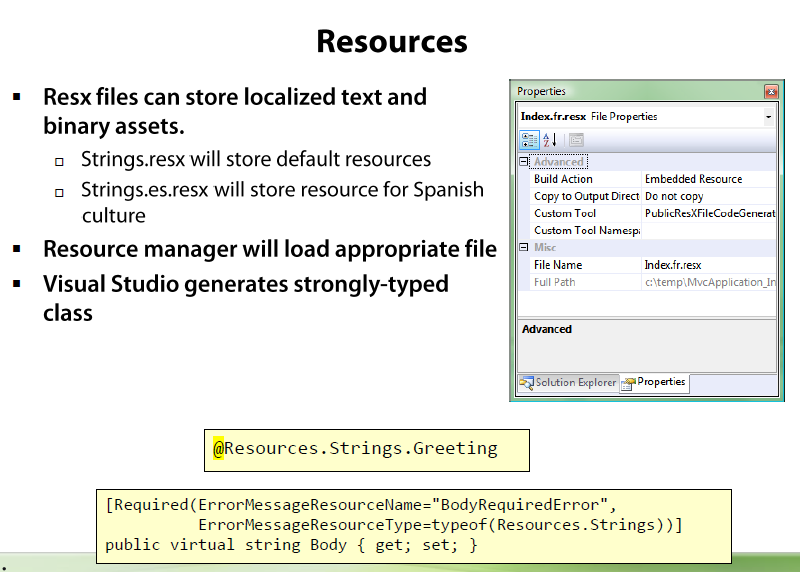




[Improving Performance with Output Caching(C#)](http://www.asp.net/mvc/tutorials/older-versions/controllers-and-routing/improving-performance-with-output-caching-cs)

# [Understanding Caching in Asp.Net MVC with example](http://www.dotnet-tricks.com/Tutorial/mvc/4R5c050113-Understanding-Caching-in-Asp.Net-MVC-with-example.html)

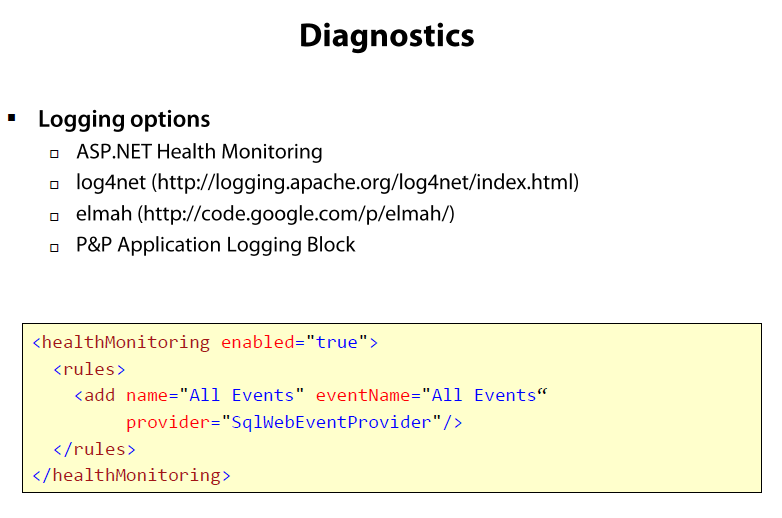




# [ASP.NET MVC Localization: Generate resource files and localized views using custom templates](http://ruijarimba.wordpress.com/2011/05/16/asp-net-mvc-localization-generate-resource-files-and-localized-views-using-custom-templates/)

# [Enabling Internationalization (i18n) in ASP.NET MVC 3 or ASP.NET MVC 4 using Resource Files](http://codingatilivedigitally.wordpress.com/2013/05/22/enabling-internationalization-i18n-in-asp-net-mvc-3-or-asp-net-mvc-4-using-resource-files/)

## [Globalization, Internationalization and Localization in ASP.NET MVC 3, JavaScript and jQuery - Part 1](http://www.hanselman.com/blog/GlobalizationInternationalizationAndLocalizationInASPNETMVC3JavaScriptAndJQueryPart1.aspx)

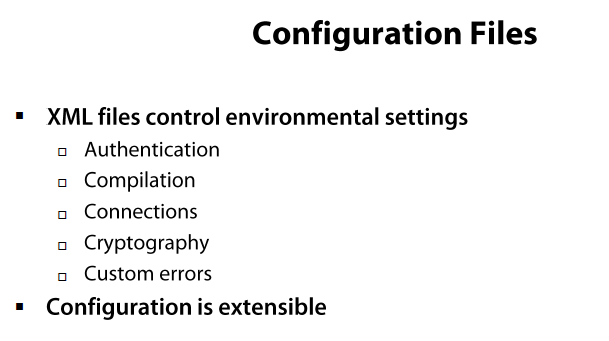


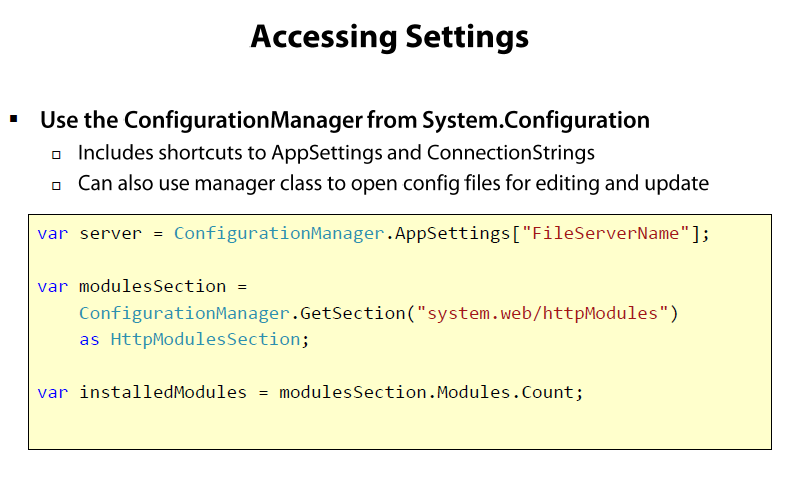
## [Log exceptions with Health Monitoring in ASP.NET MVC3](http://mattfrear.com/2011/01/28/mvc3-logging-exceptions/)

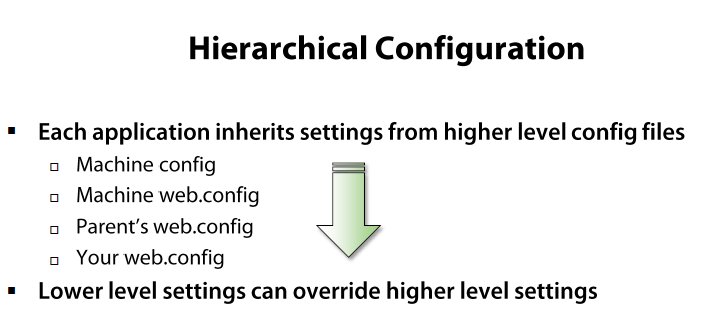
# [ASP.NET Health Monitoring Overview](http://msdn.microsoft.com/library/bb398933.ASPX)

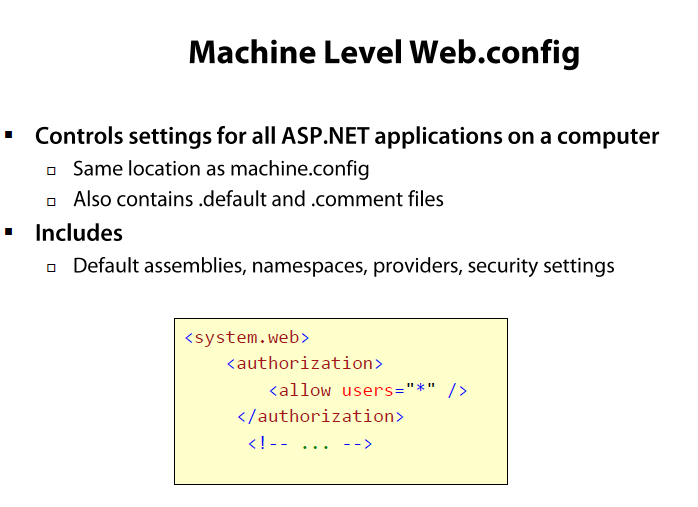
# [Health Monitoring and ASP.NET MVC](http://odetocode.com/blogs/scott/archive/2009/10/12/health-monitoring-and-asp-net-mvc.aspx)











[**How to make custom error pages work in ASP.NET MVC 4?**](http://channel9.msdn.com/Forums/Coffeehouse/How-to-make-custom-error-pages-work-in-ASPNET-MVC-4/LastRead?lastReadTime=12%2F16%2F2012%2020%3A45%3A48%20%2B00%3A00)

# [Custom Error Page with ASP.Net MVC 4](http://thirteendaysaweek.com/2012/09/25/custom-error-page-with-asp-net-mvc-4/)

# [Exception Handling in ASP.NET MVC](http://www.codeproject.com/Articles/422572/Exception-Handling-in-ASP-NET-MVC)

# [ASP.NET MVC Custom Error 404 (Not Found) Page](http://www.ssidelnikov.ru/en/2011/09/asp-net-custom-error-404-page/)



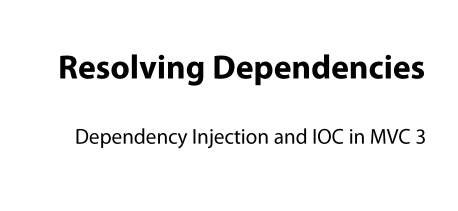
# [Deploy a Secure ASP.NET MVC 5 app with Membership, OAuth, and SQL Database to a Windows Azure Web Site](http://www.windowsazure.com/en-us/develop/net/tutorials/web-site-with-sql-database/)

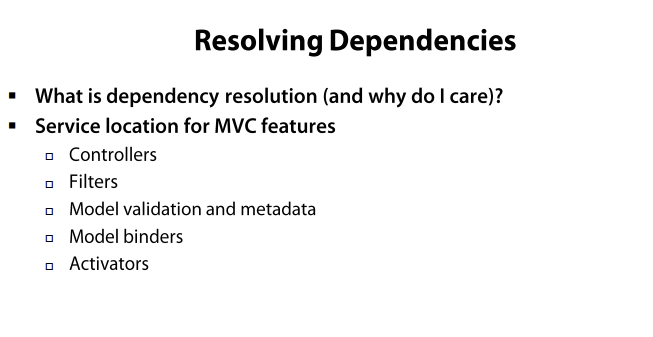
# [Testing Web Deploy publishing from Visual Studio 2010 and WebMatrix](http://www.iis.net/learn/publish/using-web-deploy/testing-web-deploy-publishing-from-visual-studio-2010-and-webmatrix)

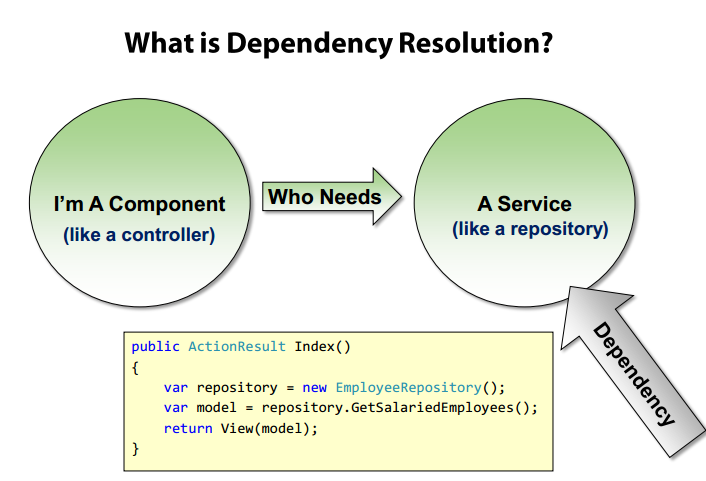
# [Deploying Application Packages](http://www.iis.net/learn/publish)

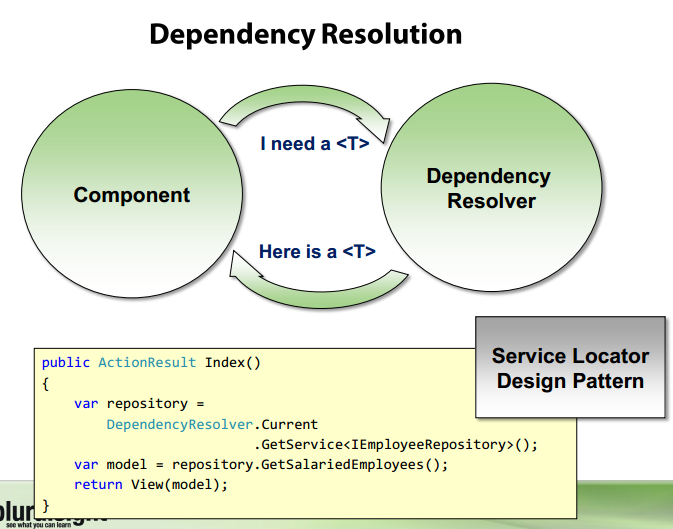
# [Building a Web Deploy Package from Visual Studio 2010](http://www.iis.net/learn/publish/using-web-deploy/building-a-web-deploy-package-from-visual-studio-2010)

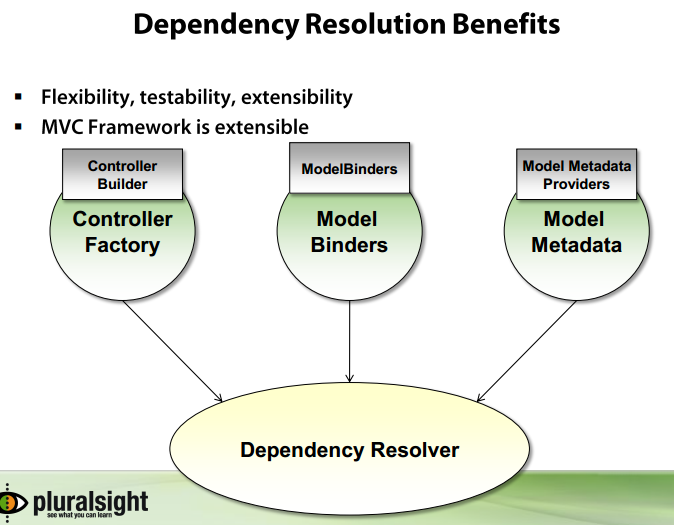
# [Bin Deploying ASP.NET MVC 3](http://haacked.com/archive/2011/05/25/bin-deploying-asp-net-mvc-3.aspx/)

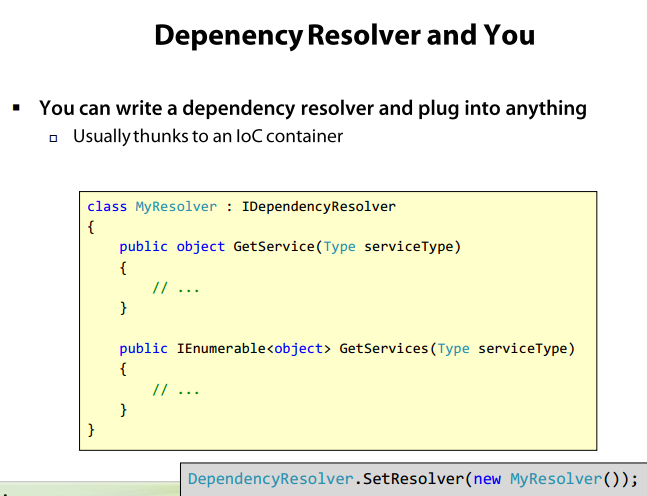


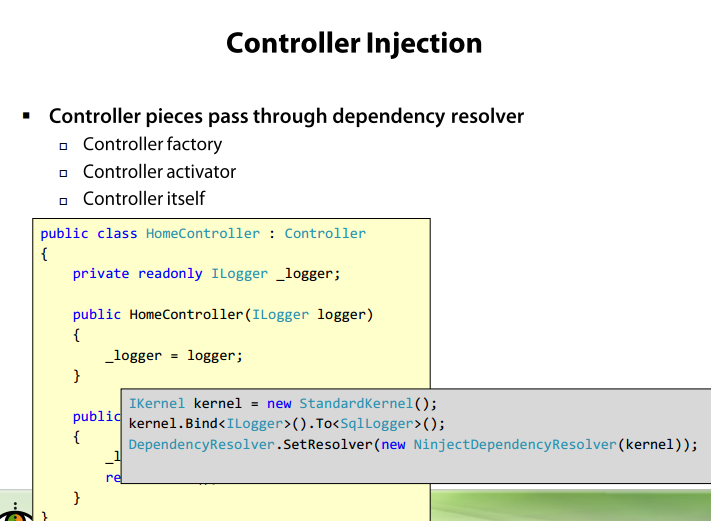


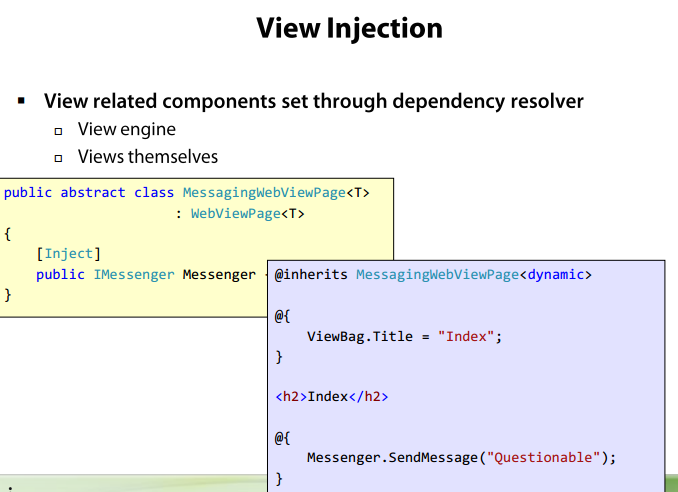




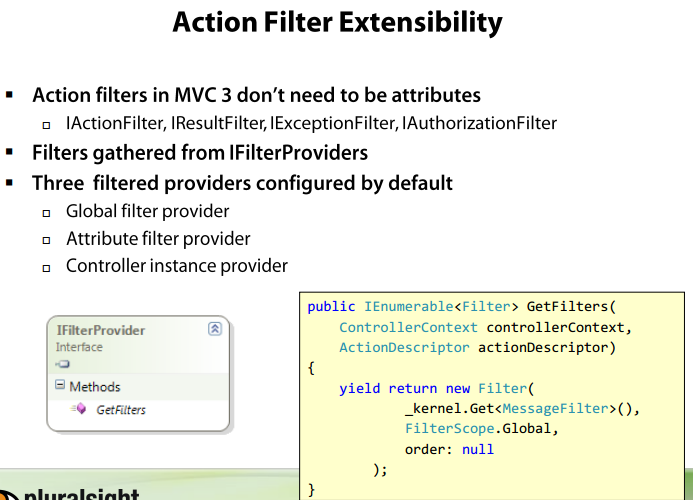


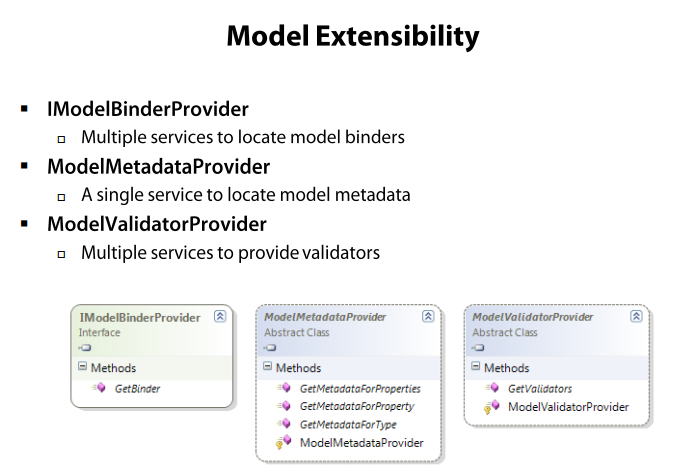


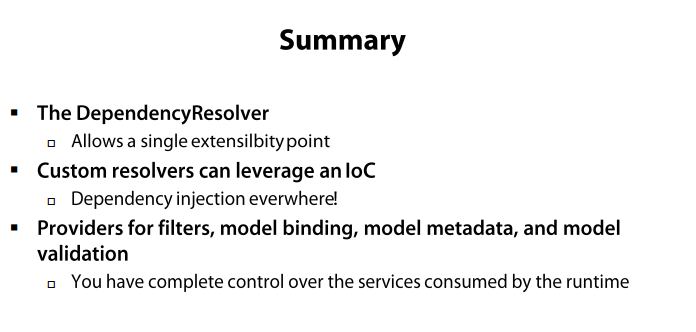












# [ASP.NET MVC Controller Dependency Injection for Beginners](http://www.codeproject.com/Articles/560798/ASP-NET-MVC-Controller-Dependency-Injection-for-Be)

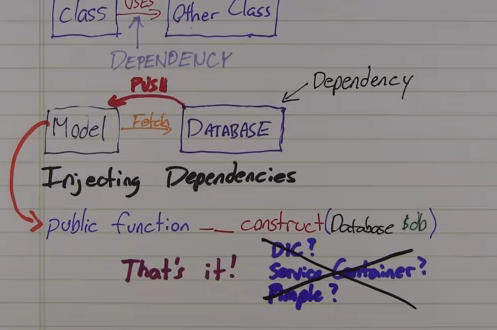
# [TDD and Dependency Injection with ASP.NET MVC](http://haacked.com/archive/2007/12/07/tdd-and-dependency-injection-with-asp.net-mvc.aspx/)

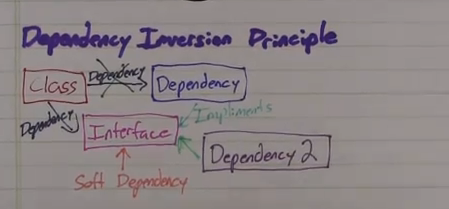
# [ASP.NET MVC 3 Dependency Injection](http://msdn.microsoft.com/en-us/gg618491)

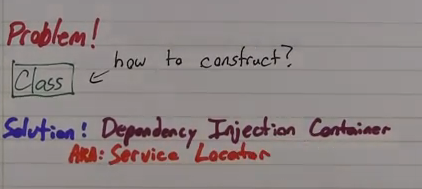
[ASP.NET MVC 4 Content Map](http://www.asp.net/mvc/overview/getting-started/aspnet-mvc-content-map)

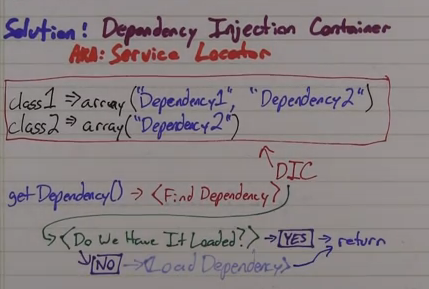
[ASP.NET MVC 4 Dependency Injection](http://www.asp.net/mvc/tutorials/hands-on-labs/aspnet-mvc-4-dependency-injection)

[Dependency Injection in SignalR](http://www.asp.net/signalr/overview/signalr-20/extensibility/dependency-injection)









# Dependency Injection

<http://www.youtube.com/watch?v=IKD2-MAkXyQ>

# Understanding Inversion of Control, Dependency Injection and Service Locator

<http://www.dotnet-tricks.com/Tutorial/dependencyinjection/bSVa100413-Understanding-Inversion-of-Control,-Dependency-Injection-and-Service-Locator.html>

# Dependency Injection in ASP.NET MVC 4 using Unity IoC Container

<http://www.dotnet-tricks.com/Tutorial/dependencyinjection/632V140413-Dependency-Injection-in-ASP.NET-MVC-4-using-Unity-IoC-Container.html>