

ASP.NET MVC Music Store Tutorial

Version 3.0b

Jon Galloway - Microsoft

4/28/2011

ASP.NET MVC Music Store Tutorial

Contents

Overview.....	4
1. File -> New Project.....	9
Installing the software.....	9
Creating a new ASP.NET MVC 3 project.....	11
2. Controllers	15
Adding a HomeController.....	15
Running the Application.....	17
Adding a StoreController.....	19
3. Views and Models.....	24
Adding a View template.....	24
Using a Layout for common site elements.....	27
Updating the StyleSheet.....	29
Using a Model to pass information to our View.....	31
Adding Links between pages.....	41
4. Data Access.....	44
Database access with Entity Framework Code-First	44
Changes to our Model Classes.....	44
Adding the App_Data folder.....	45
Creating a Connection String in the web.config file.....	46
Adding a Context Class	46
Adding our store catalog data	47
Querying the Database.....	48
Updating the Store Index to query the database.....	49
Updating Store Browse and Details to use live data	49
5. Edit Forms using Scaffolding.....	54
Creating the StoreManagerController.....	54
Modifying a Scaffolded View.....	55
A first look at the Store Manager	57

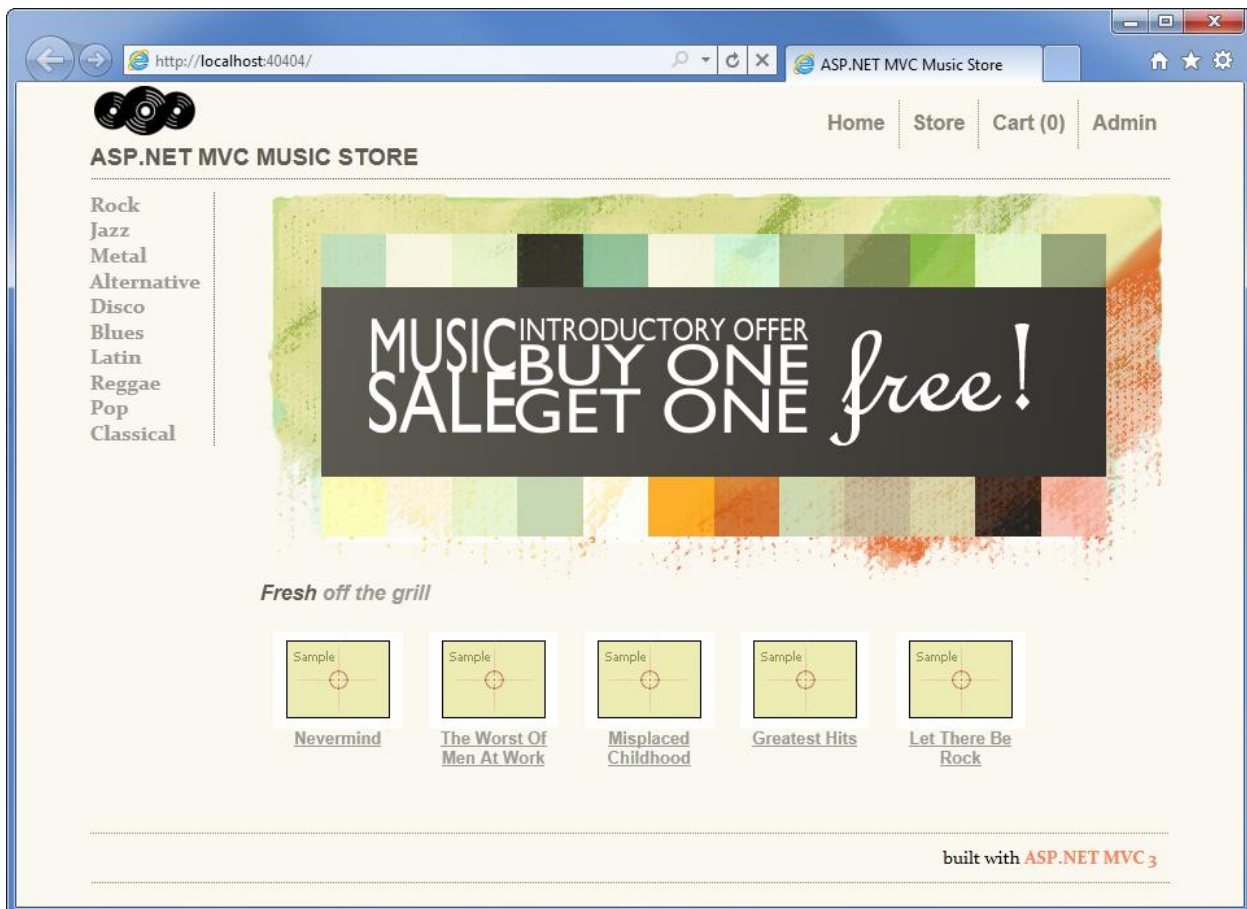
Looking at the Store Manager Controller code	61
Store Manager Index and Details actions.....	62
The Create Action Methods.....	62
Passing information to a View using ViewBag.....	62
HTML Helpers to display the Drop Downs in the Create View.....	63
Handling the Posted Form values.....	64
Handling Edits.....	66
Handling Deletion.....	68
Using a custom HTML Helper to truncate text.....	72
6. Using Data Annotations for Model Validation	76
Adding Validation to our Album Forms	76
Testing the Client-Side Validation	79
7. Membership and Authorization	81
Adding the AccountController and Views	81
Adding an Administrative User with the ASP.NET Configuration site.....	82
Role-based Authorization.....	87
8. Shopping Cart with Ajax Updates	89
Adding the Cart, Order, and OrderDetail model classes	89
Managing the Shopping Cart business logic.....	91
ViewModels.....	95
The Shopping Cart Controller	97
Ajax Updates with jQuery.....	99
9. Registration and Checkout	109
Migrating the Shopping Cart	113
Creating the CheckoutController.....	114
Adding the AddressAndPayment view	119
Defining validation rules for the Order	121
Adding the Checkout Complete view	123
Updating The Error view.....	124
10. Final updates to Navigation and Site Design.....	126
Creating the Shopping Cart Summary Partial View.....	126
Creating the Genre Menu Partial View	128

Updating Site Layout to display our Partial Views	130
Update to the Store Browse page	130
Updating the Home Page to show Top Selling Albums	132
Conclusion	135

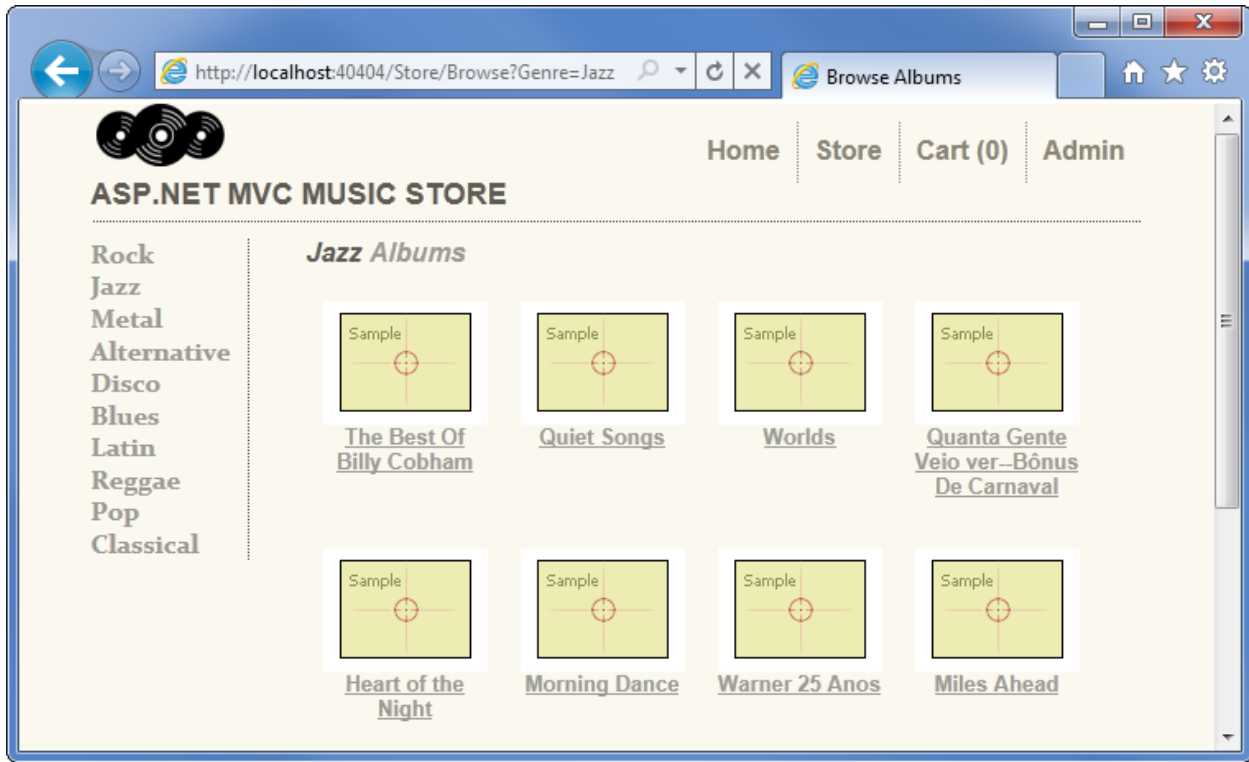
Overview

The MVC Music Store is a tutorial application that introduces and explains step-by-step how to use ASP.NET MVC and Visual Web Developer for web development. We'll be starting slowly, so beginner level web development experience is okay.

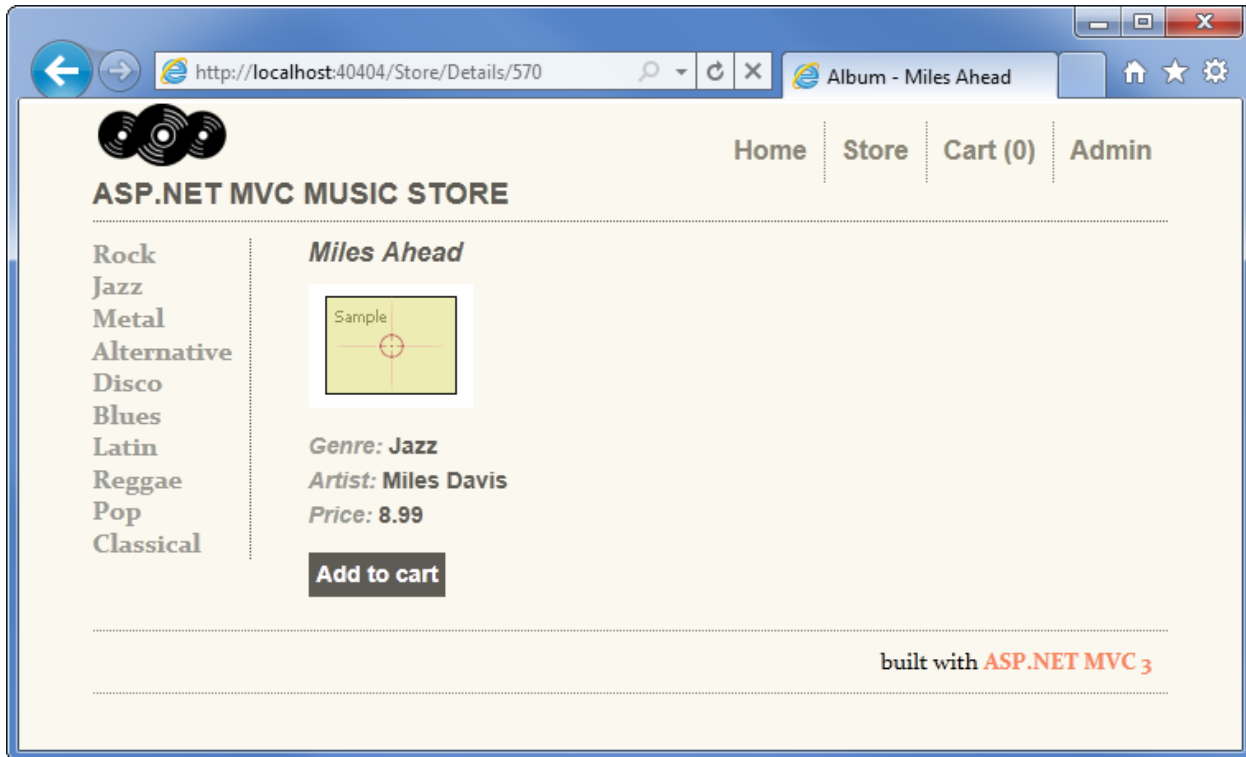
The application we'll be building is a simple music store. There are three main parts to the application: shopping, checkout, and administration.



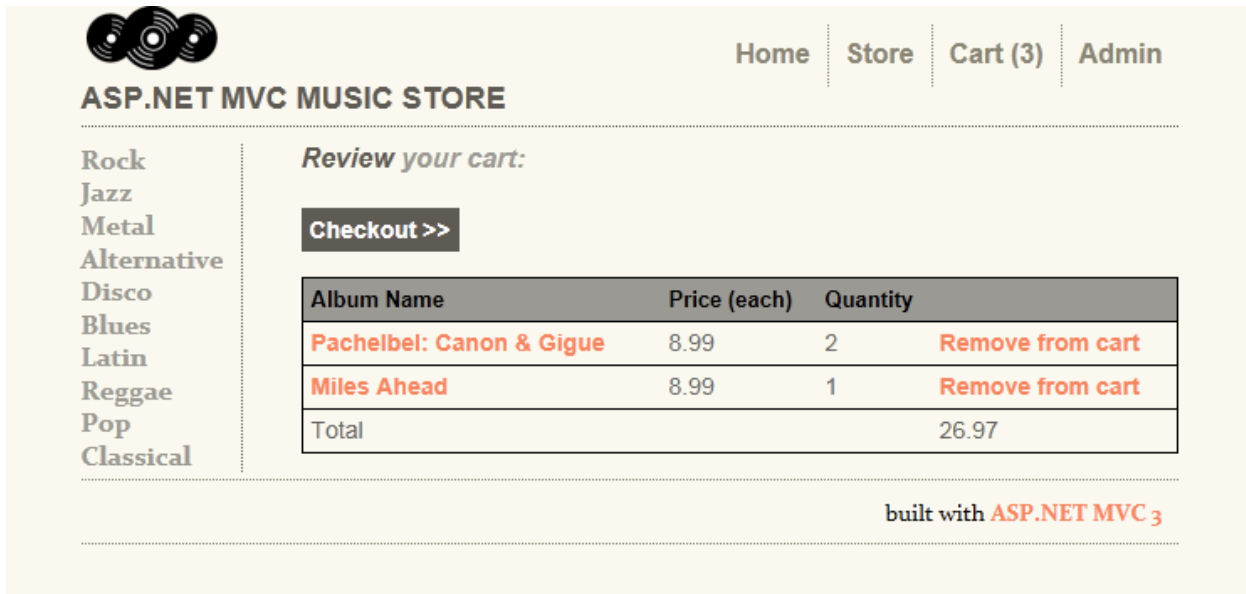
Visitors can browse Albums by Genre:



They can view a single album and add it to their cart:



They can review their cart, removing any items they no longer want:



Proceeding to Checkout will prompt them to login or register for a user account.



ASP.NET MVC MUSIC STORE

- Rock
- Jazz
- Metal
- Alternative
- Disco
- Blues
- Latin
- Reggae
- Pop
- Classical

Log On

Please enter your username and password. [Register](#) if you don't have an account.

Account Information

User name

Password

Remember me?



ASP.NET MVC MUSIC STORE

- Rock
- Jazz
- Metal
- Alternative
- Disco
- Blues
- Latin
- Reggae
- Pop
- Classical

Create a New Account

Use the form below to create a new account.

Passwords are required to be a minimum of 6 characters in length.

Account Information

User name

Email address

Password

Confirm password

After creating an account, they can complete the order by filling out shipping and payment information. To keep things simple, we're running an amazing promotion: everything's free if they enter promotion code "FREE"!

Rock
Jazz
Metal
Alternative
Disco
Blues
Latin
Reggae
Pop
Classical

Shipping Information

First Name

Last Name

Address

City

State

Postal Code

Country

Phone

Email Address

Payment

We're running a promotion: all music is free with the promo code "FREE"

Promo Code

After ordering, they see a simple confirmation screen:



[Home](#) | [Store](#) | [Cart \(0\)](#) | [Admin](#)

ASP.NET MVC MUSIC STORE

Rock
Jazz
Metal
Alternative
Disco
Blues
Latin
Reggae
Pop
Classical

Checkout Complete

Thanks for your order! Your order number is: 475

How about shopping for some more music in our [store](#)

built with **ASP.NET MVC 3**

In addition to customer-facing pages, we'll also build an administrator section that shows a list of albums from which Administrators can Create, Edit, and Delete albums:

Albums

Create New Album

	Title	Artist	Genre
Edit Delete	For Those About To Rock W...	AC/DC	Rock
Edit Delete	Let There Be Rock	AC/DC	Rock
Edit Delete	Greatest Hits	Lenny Kravitz	Rock
Edit Delete	Misplaced Childhood	Marillion	Rock
Edit Delete	The Best Of Men At Work	Men At Work	Rock
Edit Delete	Nevermind	Nirvana	Rock
Edit Delete	Compositores	O Terço	Rock
Edit Delete	Bark at the Moon (Remaste...	Ozzy Osbourne	Rock

1. File -> New Project

Installing the software

This tutorial will begin by creating a new ASP.NET MVC 3 project using the free Visual Web Developer 2010 Express (which is free), and then we'll incrementally add features to create a complete functioning application. Along the way, we'll cover database access, form posting scenarios, data validation, using master pages for consistent page layout, using AJAX for page updates and validation, user login, and more.

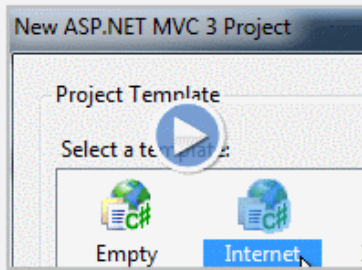
You can follow along step by step, or you can download the completed application from <http://mvcmusicstore.codeplex.com>.

You can use either Visual Studio 2010 SP1 or Visual Web Developer 2010 Express SP1 (a free version of Visual Studio 2010) to build the application. We'll be using the SQL Server Compact (also free) to host the database. Before you start, make sure you've installed the prerequisites listed below. You can install all of them using the following Web Platform Installer

link: <http://www.microsoft.com/web/gallery/install.aspx?appid=VWD2010SP1Pack>

Note: You can find this link on the big green button at this (easier to remember) link: <http://asp.net/mvc>

What is ASP.NET MVC?



ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and that gives you full control over markup for enjoyable, agile development. ASP.NET MVC includes many features that enable fast, TDD-friendly development for creating sophisticated applications that use the latest web standards.

Visual Studio Express provides a free development tool that makes MVC development easy.



Already have Visual Studio? Just download the [MVC 3 installer](#).

The Web Platform Installer will check what you've got installed and just download what you need.



If you want to individually install the prerequisites using the following links instead of using the above link, use the following links (written out in case you're using a printed version of this tutorial):

- [Visual Studio Web Developer Express SP1 prerequisites](http://www.microsoft.com/web/gallery/install.aspx?appid=VWD2010SP1Pack)
http://www.microsoft.com/web/gallery/install.aspx?appid=VWD2010SP1Pack
- [ASP.NET MVC 3 Tools Update](http://www.microsoft.com/web/gallery/install.aspx?appid=MVC3)
http://www.microsoft.com/web/gallery/install.aspx?appid=MVC3
- [SQL Server Compact 4.0](http://www.microsoft.com/web/gallery/install.aspx?appid=SQLCE;SQLCEVSTools_4_0) - including both runtime and tools support
http://www.microsoft.com/web/gallery/install.aspx?appid=SQLCE;SQLCEVSTools_4_0

Note: If you're using Visual Studio 2010 instead of Visual Web Developer 2010, install the prerequisites with this link instead:

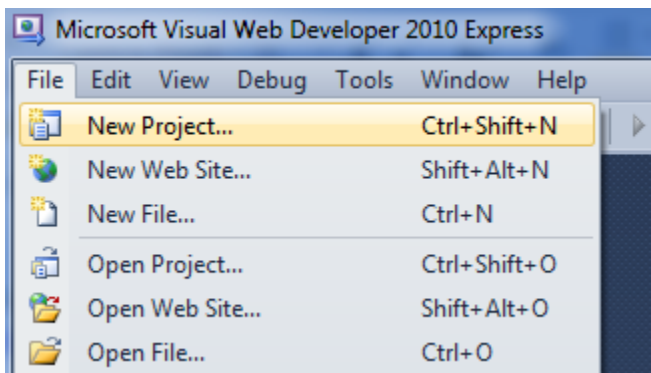
[Visual Studio Web Developer Express SP1 prerequisites](http://www.microsoft.com/web/gallery/install.aspx?appxml=&appid=VS2010SP1Pack)

http://www.microsoft.com/web/gallery/install.aspx?appxml=&appid=VS2010SP1Pack

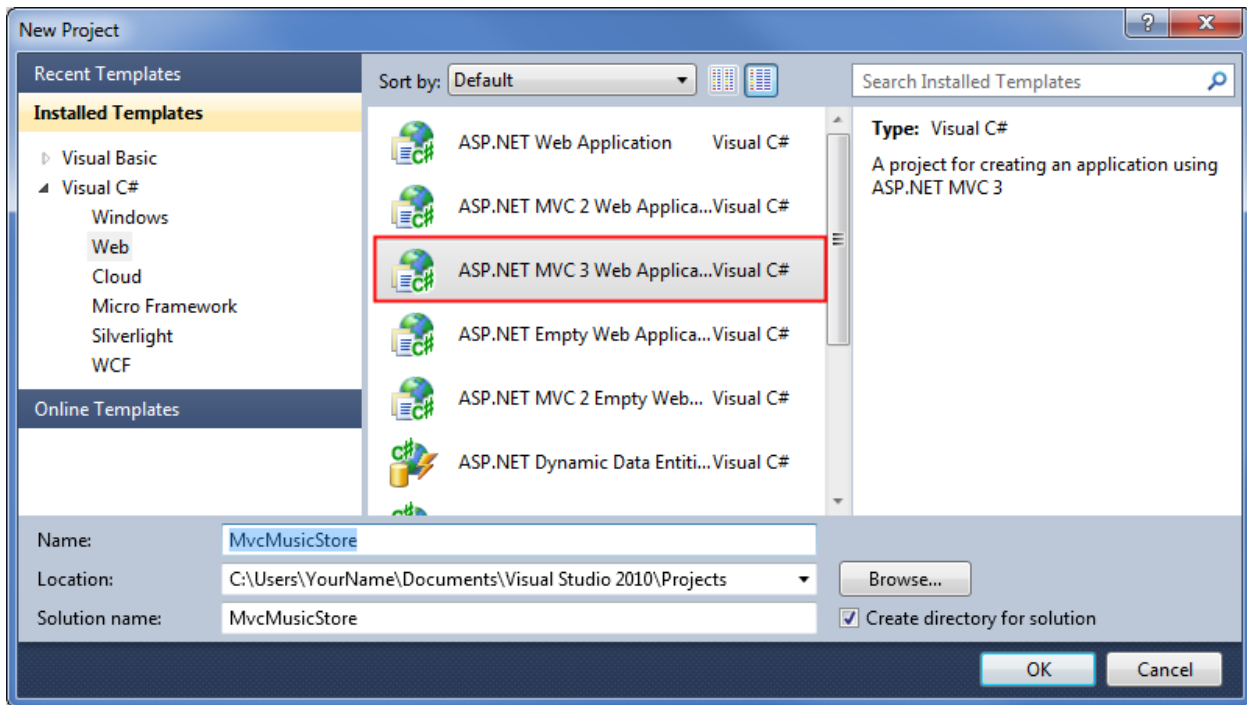
I highly recommend you use the first Web Platform Installer link, as it will make sure you've got everything set up correctly.

Creating a new ASP.NET MVC 3 project

We'll start by selecting "New Project" from the File menu in Visual Web Developer. This brings up the New Project dialog.



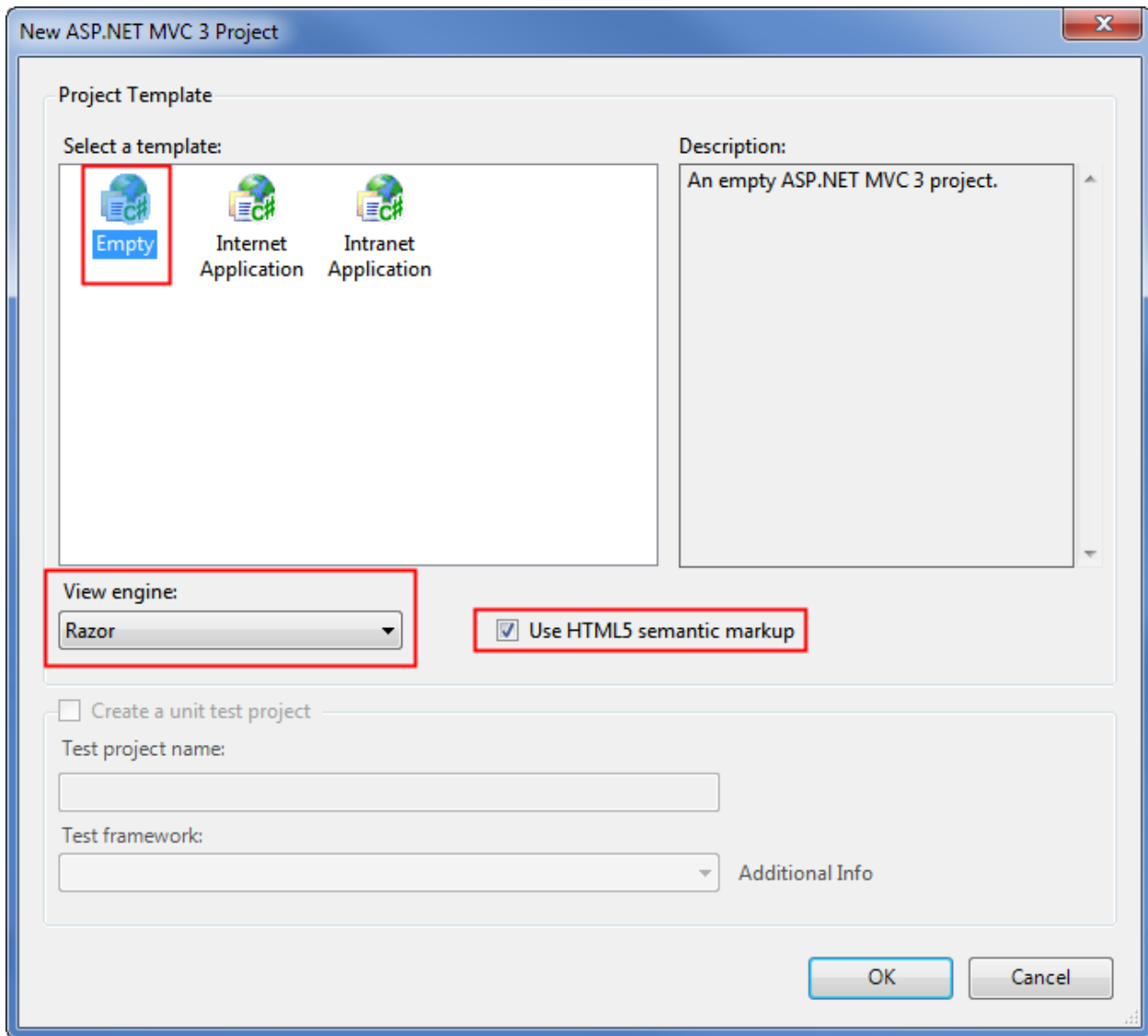
We'll select the Visual C# -> Web Templates group on the left, then choose the "ASP.NET MVC 3 Web Application" template in the center column. Name your project MvcMusicStore and press the OK button.



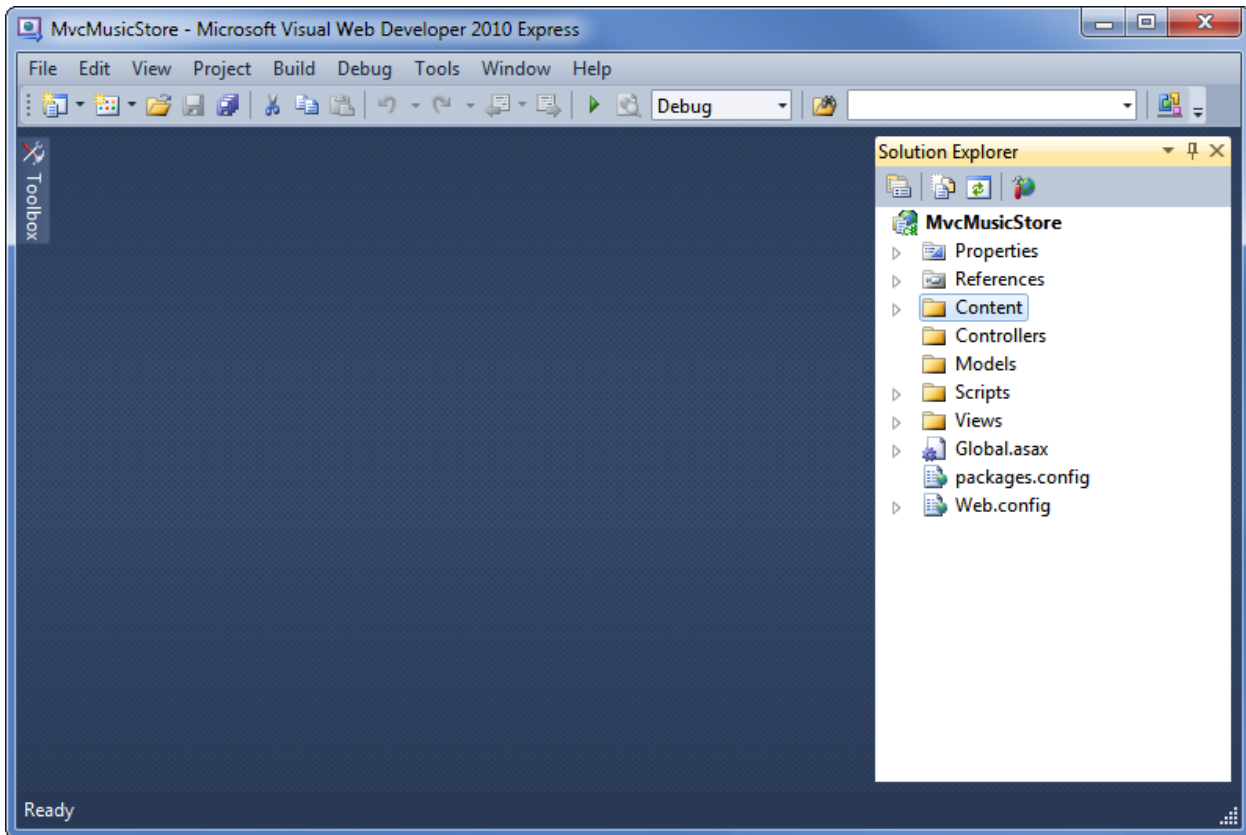
This will display a secondary dialog which allows us to make some MVC specific settings for our project. Select the following:

- Project Template - select Empty
- View Engine - select Razor
- Use HTML5 semantic markup - checked

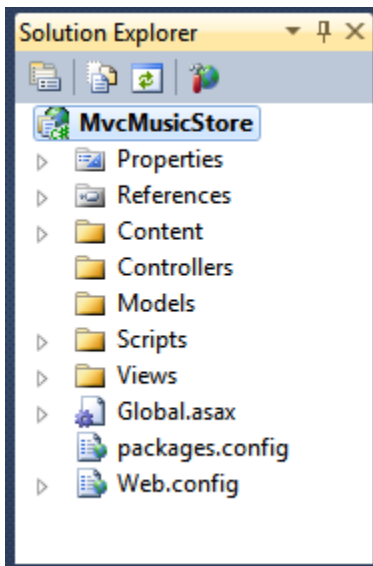
Verify that your settings are as shown below, then press the OK button.



This will create our project. Let's take a look at the folders that have been added to our application in the Solution Explorer on the right side.



The Empty MVC 3 template isn't completely empty – it adds a basic folder structure:



ASP.NET MVC makes use of some basic naming conventions for folder names:

Folder	Purpose
/Controllers	Controllers respond to input from the browser, decide what to do with it, and return response to the user.
/Views	Views hold our UI templates

/Models	Models hold and manipulate data
/Content	This folder holds our images, CSS, and any other static content
/Scripts	This folder holds our JavaScript files

These folders are included even in an Empty ASP.NET MVC application because the ASP.NET MVC framework by default uses a “convention over configuration” approach and makes some default assumptions based on folder naming conventions. For instance, controllers look for views in the Views folder by default without you having to explicitly specify this in your code. Sticking with the default conventions reduces the amount of code you need to write, and can also make it easier for other developers to understand your project. We’ll explain these conventions more as we build our application.

2. Controllers

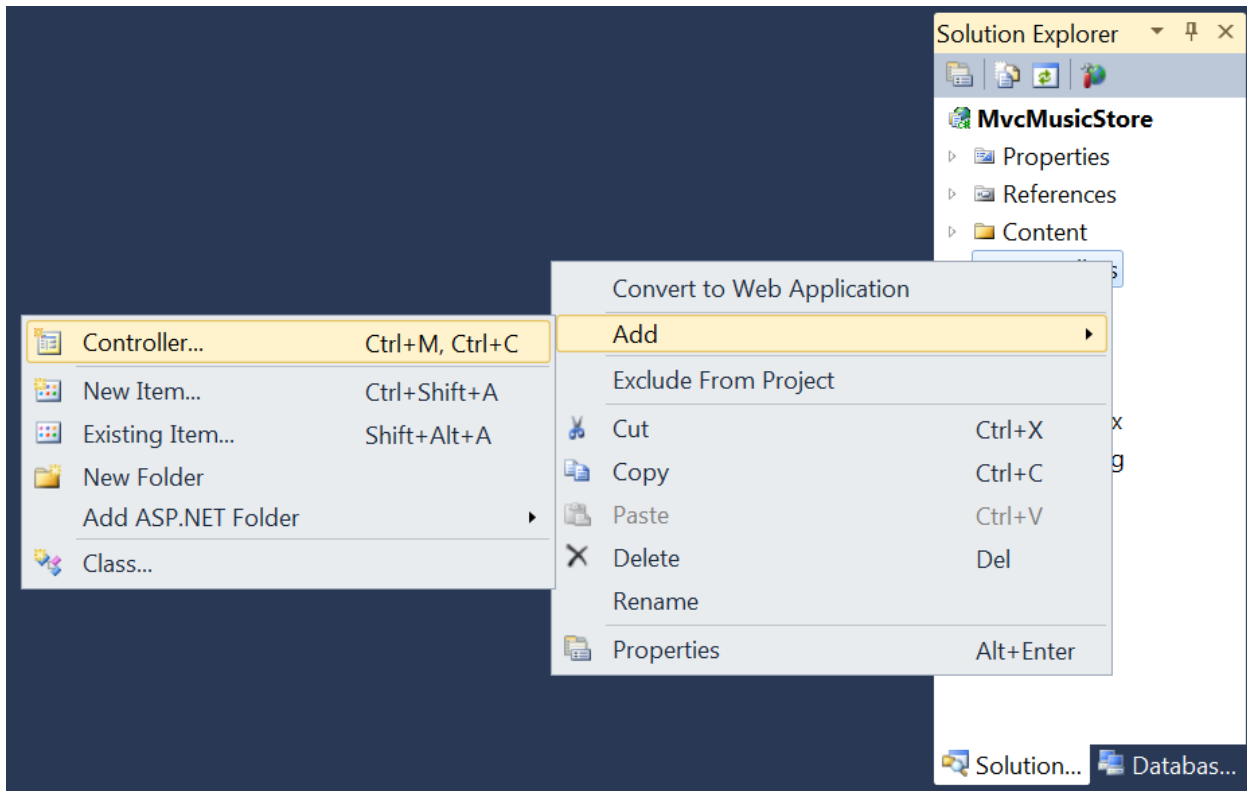
With traditional web frameworks, incoming URLs are typically mapped to files on disk. For example: a request for a URL like `"/Products.aspx"` or `"/Products.php"` might be processed by a `"Products.aspx"` or `"Products.php"` file.

Web-based MVC frameworks map URLs to server code in a slightly different way. Instead of mapping incoming URLs to files, they instead map URLs to methods on classes. These classes are called `"Controllers"` and they are responsible for processing incoming HTTP requests, handling user input, retrieving and saving data, and determining the response to send back to the client (display HTML, download a file, redirect to a different URL, etc.).

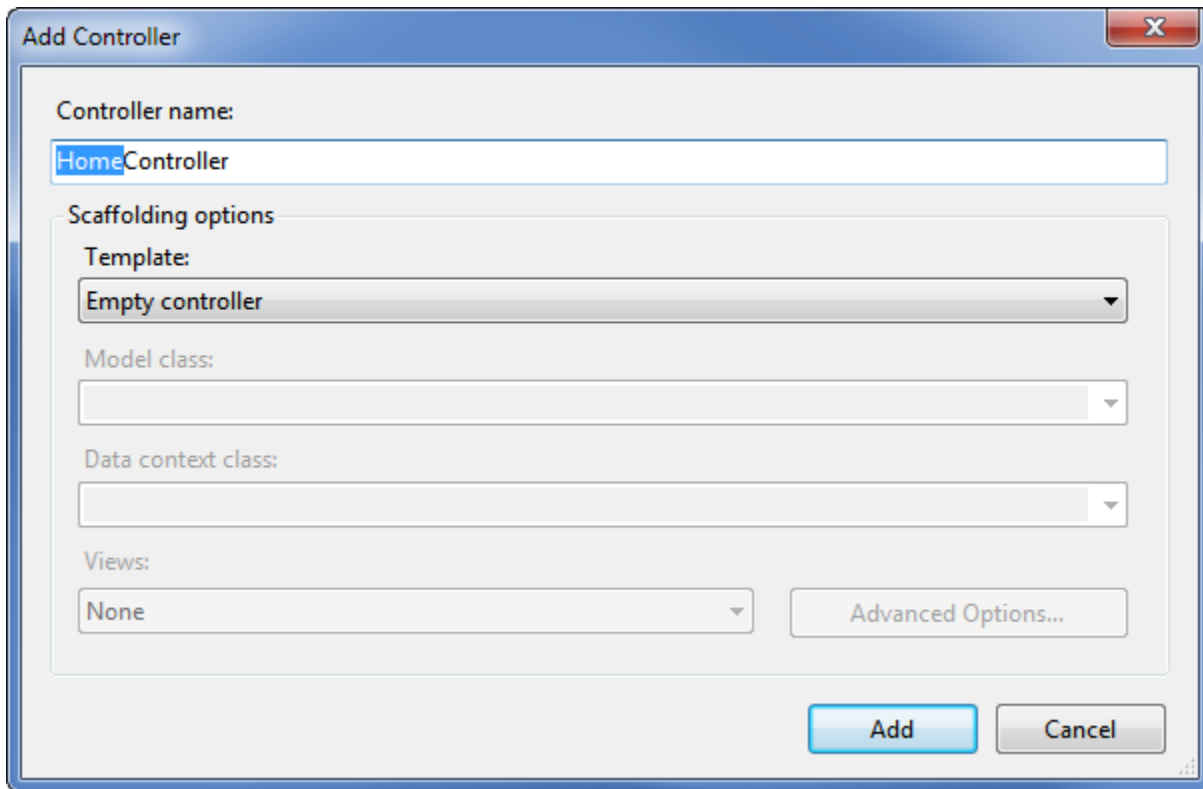
Adding a HomeController

We’ll begin our MVC Music Store application by adding a Controller class that will handle URLs to the Home page of our site. We’ll follow the default naming conventions of ASP.NET MVC and call it `HomeController`.

Right-click the `"Controllers"` folder within the Solution Explorer and select `"Add"`, and then the `"Controller..."` command:



This will bring up the “Add Controller” dialog. Name the controller “HomeController” and press the Add button.



Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

