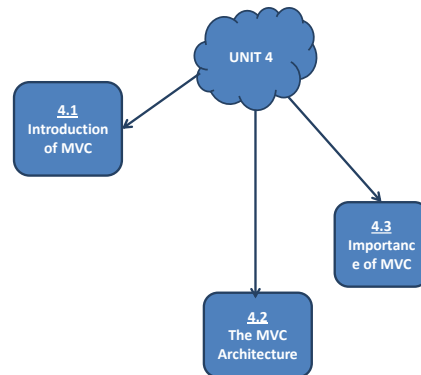


Introduction to MVC

- UNIT 4



4.1 Introduction of MVC

- What is MVC?
- Road to MVC4
- MVC4
- ASP.NET Web API
- Default Project Template

What is MVC?

- Model-View-Controller (MVC) has been an important architectural pattern in computer science for many years.
- Originally named *Thing-Model-View-Editor* in 1979, it was later simplified to *Model-View-Controller*.
- It is a powerful and elegant means of separating concerns within an application (for example, separating data access logic from display logic) and applies itself extremely well to web applications.
- Its explicit separation of concerns does add a small amount of extra complexity to an application's design, but the extraordinary benefits outweigh the extra effort.

Con..

- The MVC pattern is used frequently in web programming. With ASP.NET MVC, it's translated roughly as:
 - **Models:** These are the classes that represent the domain you are interested in. These domain objects often encapsulate data stored in a database as well as code used to manipulate the data and enforce domain-specific business logic. With ASP.NET MVC, this is most likely a Data Access Layer of some kind, using a tool like Entity Framework or NHibernate combined with custom code containing domain-specific logic.
 - **View:** This is a template to dynamically generate HTML.
 - **Controller:** This is a special class that manages the relationship between the View and the Model. It responds to user input, talks to the Model, and decides which view to render (if any). In ASP.NET MVC, this class is conventionally denoted by the suffix *Controller*.

Road to MVC4

- In the three short years since ASP.NET MVC 1 was released in March 2009, we've seen four major releases of ASP.NET MVC and several more interim releases.
 - ASP.NET MVC 1 Overview
 - ASP.NET MVC 2 Overview
 - ASP.NET MVC 3 Overview
 - ASP.NET MVC 4 Overview

ASP.NET MVC 1 Overview

- In February 2007, Scott Guthrie (“ScottGu”) of Microsoft sketched out the core of ASP.NET MVC.
- The official MVC 1.0 release — including code and unit tests — on 13 March 2009.

ASP.NET MVC 2 Overview

- ASP.NET MVC 2 was released just one year later, in March 2010. Some of the main features in MVC 2 included:
 - UI helpers with automatic scaffolding with customizable templates
 - Attribute-based model validation on both client and server
 - Strongly typed HTML helpers
 - Improved Visual Studio tooling
- There were also lots of API enhancements and “pro” features, based on feedback from developers building a variety of applications on ASP.NET MVC 1, such as:
 - Support for partitioning large applications into *areas*
 - Asynchronous controllers support
 - Support for rendering subsections of a page/site using `Html.RenderAction`
 - Lots of new helper functions, utilities, and API enhancements

ASP.NET MVC 3 Overview

- ASP.NET MVC 3 shipped just 10 months after MVC 2, driven by the release date for Web Matrix. Some of the top features in MVC 3 included:
 - The Razor view engine
 - Support for .NET 4 Data Annotations
 - Improved model validation
 - Greater control and flexibility with support for dependency resolution and global action filters
 - Better JavaScript support with unobtrusive JavaScript, jQuery Validation, and JSON binding

ASP.NET MVC 4 Overview

- The MVC 4 release is building on a pretty mature base and is able to focus on some more advanced scenarios. Some top features include:
 - ASP.NET Web API
 - Enhancements to default project templates
 - Mobile project template using jQuery Mobile
 - Display Modes
 - Task support for Asynchronous Controllers
 - Bundling and minification

ASP.NET Web API

- ASP.NET Web API (referred to as *Web API*), a framework that offers the ASP.NET MVC development style but is tailored to writing HTTP services. This includes both modifying some ASP.NET MVC concepts to the HTTP service domain and supplying some new service-oriented features.

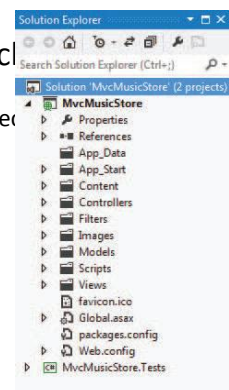
1. Routing

2. Model binding and validation

3. Filters

4.2 The MVC Architecture

- File and directories of Project



Con..

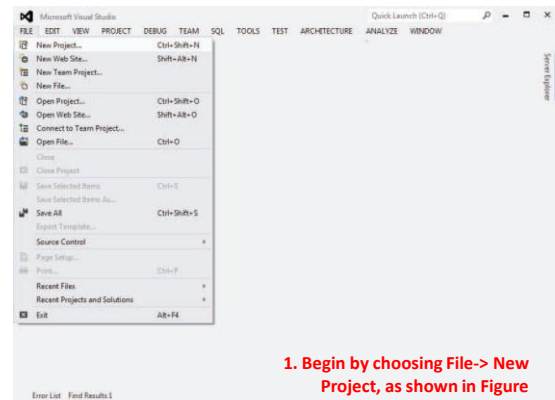
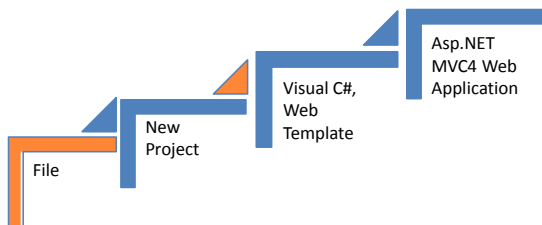
DIRECTORY	PURPOSE
/Controllers	Where you put Controller classes that handle URL requests
/Models	Where you put classes that represent and manipulate data and business objects
/Views	Where you put UI template files that are responsible for rendering output, such as HTML
/Scripts	Where you put JavaScript library files and scripts (.js)
/Images	Where you put images used in your site
/Content	Where you put CSS and other site content, other than scripts and images
/Filters	Where you put filter code. Filters are an advanced feature,
/App_Data	Where you store data files you want to read/write
/App_Start	Where you put configuration code for features like Routing, Bundling, and Web API

4.3 Importance of MVC

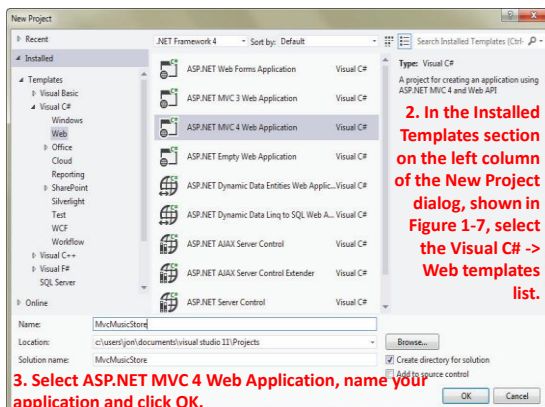
- Configuration logic moved to App_Start
- Empty MVC project Template
- Add Controller anywhere
- Asynchronous controller
- Display modes
- Bundling and minification
- Web API

Creating MVC based Application

- Steps



1. Begin by choosing File-> New Project, as shown in Figure



3. Select ASP.NET MVC 4 Web Application, name your application and click OK.