WEB524

WEB PROGRAMMING ON WINDOWS

WEEK 1 – LECTURE 1

INTRODUCTION TO ASP.NET MVC

Resources

Read these chapters in the textbook:

- Foreword
- Introduction
- Chapter 1 – Getting Started
- Chapter 2 – Controllers (skim the content)
- Chapter 3 – Views (skim the content)

Optional Readings

- ASP.NET MVC 1 Overview to ASP.NET MVC 4 Overview (Page 4-7)
- ASP.NET Web API (Pages 7 to 9); we'll revisit this later in the course

Notes

- On page 16, in the "Installing ASP.NET MVC 5" section, the textbook states that "MVC 5 is included with Visual Studio 2013, so there's nothing to install". This also holds true for Visual Studio 2015 to Visual Studio 2019 – ASP.NET MVC is included and built in.

What is a web app?

- A ‘web app’ is a program that runs on a web server.
- A web server uses the HTTP application protocol to communicate with users.
- A ‘web browser’ is typically used to access a web app.
- Responses from a web app are typically composed of HTML however other internet media types may make up the payload of a request or response.

Web App Development

- In the last decade or so, two web app dev frameworks have dominated:
  - ASP.NET
  - PHP
- Other than these two, you'll see some Ruby on Rails and Java solutions (e.g. Spring) out there. In the future, the trend is leaning towards full-stack JavaScript.
- The Model-View-Controller (MVC) software design pattern is now the de facto development approach for web apps.
- Both ASP.NET and PHP offer the ability to use the MVC pattern.

ASP.NET Web Forms

- ASP.NET 1.0 was first released in 2002. It included the whole “Web Forms” stack.
- Programmers drag-and-drop server controls to the page and ASP.NET will handle the complexity of adding statefulness to a stateless architecture.
- Leads to a confusing page lifecycle, bloated HTML, and difficulties customizing.

What is an ASP.NET Application?

The following was adapted from this MSDN reference document.

- An ASP.NET application is a collection of source code files and resources (e.g. images) within a specific directory, and its subdirectories, on a single web server.
- The first time a browser/client requests a URL in an ASP.NET application, the web server creates an instance of an HttpApplicationState class.
- Application state is not shared across either a Web farm or a Web garden.
- The lifetime of the instance is twenty (20) minutes, by default (but it can be changed). The lifetime is reset with each request to a URL in the app.
- A separate single instance is created for each ASP.NET application on a Web server.
- A reference to this instance is available within the app via the ApplicationInstance property. It is a property of the HttpContext class.

The Global.asax.cs File

- The Global.asax.cs source code file defines the "MvcApplication" class, a subclass of System.Web.HttpApplication.
- You can add code to this class, to initialize software components and data for the app, and to customize behaviour when servicing a request.
- The ApplicationInstance property refers to the app's code and execution environment, whereas …
- The Application property refers to the state of the app, including its data.
- In order to use the Application property in source code files, you must get a reference to the application by accessing the System.Web.HttpContext.Application object.
  - The System.Web namespace is typically added in a using directive, so we can simply use "HttpContext.Aplication".

What is an application domain?

- Also known as the problem domain.
- An application domain (often shortened to “app domain”) is the area of knowledge that is implemented in a software application.
- In an ASP.NET MVC application, an app domain is materialized using standard software components, such as models, views, and controllers. It will include software that implements business logic and rules, workflow, and persistence management.
- Wikipedia has useful articles on problem domain, domain model, and conceptual model.

ASP.NET MVC

- ASP.NET is a runtime environment for web apps that run on the Microsoft Web Platform.
- ASP.NET MVC is a framework for building web apps that conform to the model-view-controller (MVC) design pattern.
- Two key items new developers need to know about ASP.NET MVC:
  - It uses the front controller pattern.
  - The application development process relies on strict programming conventions.

How does the front controller pattern work?

1. When you request a resource in an ASP.NET MVC app, the ASP.NET runtime receives the request.
2. A routing module inspects the URL and will determine what happens next.
3. The request is typically routed to a specific method (i.e. a function) in a specific “controller” class (which may use data in the “model“).
4. The method generates some data and passes it to a “view” for rendering by a view engine (typically Razor).

MVC - Controller

"A set of classes that handle communication from the user, overall application flow, and application-specific logic."

- The conductors of an MVC application, orchestrating the interactions of the user, models, and the views.
- An incoming request from a user is routed to a specific method in the controller. This method is referred to as an Action.
- Data (user input) is passed into the model, processed, and returned to a view for rendering.

MVC - View

“Responsible for providing the user interface (UI) to the user.”

- A view is a source code file that contains user interface code.
- The user interface target is a web browser, so a view contains HTML markup and code expressions that place data into the markup.
- After the controller has executed the appropriate logic for a request, it delegates the display to the view.
- Unlike file-based web frameworks like PHP, views are not themselves directly accessible. You can’t point your browser to a view to see the contents.
- Not all views will render HTML but it the most common format.
- Not all views require data from the controller to render.

MVC - Model

"A set of classes that describes the data you're working with, as well as the business rules for how the data can be changed and manipulated."

- A model holds and manages your app's data.
- We write classes that model the data.
- Data that's persisted in a store is modelled by design model classes.
- Data that is delivered to the user, or gathered from the user, is modelled by view model classes.
- View models are important and will be used extensively in this class.

Demo

- We will familiarize ourselves with Visual Studio 2019
  - Creating projects
  - Solution Explorer
  - NuGet Package Manager (UI)
  - Error List
  - Output Window
  - Breakpoints
  - Autos, Locals, and Watch Windows
  - Immediate Window
  - Package Manager Console
  - Run the project – Discuss IIS
- We will create a new ASP.NET MVC Web Application
  - Inspect the MvcApplication
  - Discuss routing
  - Look at controllers, views and models
  - Global.asax
  - Web.config