Asp.Net Core MVC

Asp.Net Core

- Web development framework from Microsoft that works based on .NET Core Framework.
- Asp.Net supports 3 methods of application development.
  - Asp.Net MVC
  - Asp.Net Razor Pages
  - Asp.Net Web API

Asp.Net MVC

- ASP.Net application following MVC pattern

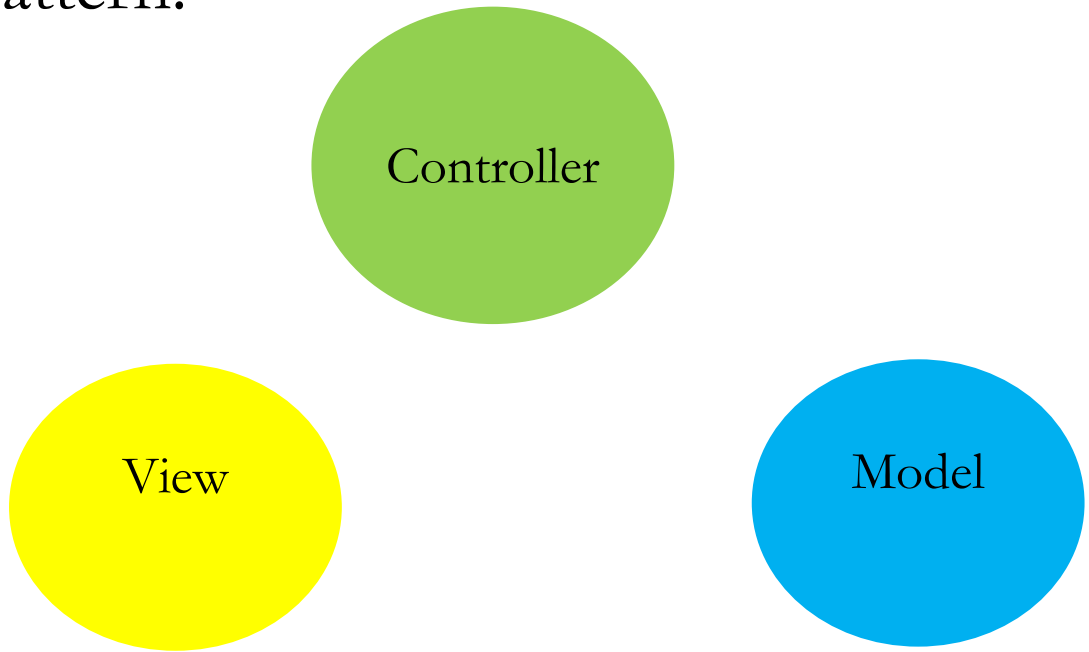
Asp.Net WebAPI

- Is used for creating RESTful APIs using ASP.Net.

Asp.Net MVC and Asp.Net Web API

MVC

- Model-View-Controller (MVC).
- Standard Architectural Pattern.
- Separation of concerns.



Web API

- Will not have the View Component
- Web API is used to provide services through http protocol.

Creating First Asp.Net Application

Creating a Project
Examine project ‘s .csproj file
Add NugetPackages

Running ASP.NET Core Applications

- Prior to .Net Core, when Asp.Net under DotNet Framework was used applications could run only from IIS webserver.
- With ASP.NET Core, applications typically run using the Kestrel web server with an option to use IIS, Apache, Nginx, etc., by way of a reverse proxy between Kestrel and the other web server.
- When application runs in development environment, the application's ports and environment are configured with a file named launchsettings.json located under the Properties folder.

NuGet

What is NuGet?

- NuGet (pronounced "New Get") is a package manager for .NET
- It is designed to enable developers to share reusable code.

Adding a Package from NuGet

- You can add packages to your project using Visual Studio's NuGet Package Manager

or

- using the command line with the command

dotnet **add** **package** *package-name*

Asp.Net Convention over Configuration

Asp.Net Convention over Configuration

- ASP.NET MVC and ASP.NET Web API reduced the amount of configuration necessary by introducing certain conventions.
- These convention's could be followed by the programmers or could be overriden
- The main conventions are naming conventions and directory structure

Naming Conventions

- Controllers are typically named with the "Controller" suffix (e.g. HomeController.
- When accessed through routing, the "Controller" suffix is dropped.
- When looking for a controller's views, the controller name minus the suffix is the starting search location.
- Another naming convention is for Razor Views.
- By default, an action method will render the view of the same name as the action method.

Naming Conventions

- Another naming convention is used in locating the views for a controller's action methods.
  - When looking for a controller's views, the controller name minus the suffix is the starting search location.
- By default, an action method will render the view of the same name as the method.

Directory Structure

- There are several folder conventions

Controllers Folder

- Is the standard folder where Controllers should come for an ASP.NET Core MVC and API applications.

Directory Structure

- There are several folder conventions

Views Folder

- The Views folder is where the views for the application are stored.
- Each controller gets its own folder under
- the main Views folder named after the controller name (minus the Controller suffix).
- The action methods will render views from their controller's folder by default.

Directory Structure

- There are several folder conventions

Shared Folder

- A special folder under Views is named Shared.
- This folder is accessible to all controllers and their action methods.
- After searching the folder named for the controller, if the view can't be found, then the Shared folder is searched for the view.

Directory Structure

- There are several folder conventions

wwwroot Folder

- This folder contains static content used in the website, such as CSS for styling, images, JavaScript, and a favicon.ico file

Standard Files

- appsettings.json contains settings that your website can load at runtime, for example, the database connection string
- .csproj : This file contains project settings like version of DotNet SDK and a list of NuGet packages that your project requires.

Standard Files

- Program.cs: This file defines a class that contains the Main entry point to the application.
- This class configures the web server used for the application.

Startup.cs

- This file adds and configures services that your website needs, for example data base, routes etc.
- Startpup.cs file contains a ConfigureServices method and a Configure method
- ConfigureServices method is used to specify the services used by the application
- and Configure method defines the pipeline for processing incoming HTTP requests