

Asp.Net MVC Conventions

Jim Fawcett

CSE686 – Internet Programming

Spring 2019

Asp.Net Conventions

- The MVC framework uses three kinds of conventions:
 - Structure conventions
 - Model structure
 - Directory structure
 - Naming conventions
 - Controller methods & views
 - Placement conventions

Model Structure

- Models are C# classes
 - All models reside in ./Models folder
 - Have public properties
 - Controller methods may name a model
 - When controller method is invoked it is passed an instance of model named as its method argument
 - Views can be typed to use a specific model
 - Model's public properties are populated on postback if their names match names used in the view.

Naming Conventions

- Default controller is HomeController.cs
 - Controller names end in Controller
 - Default controller defined in Startup.Configure method
- Default view is Index.cshtml
- Controller methods are names of views
 - `Public ActionResult Index() { ... }`

Placement Conventions

- Controllers are placed in folder Controllers
- Views are placed in subfolder named with controller name:
 `./Views/Home/Index.cshtml`
- All models are placed in folder named Models

Adding a New Controller

- Right-click on Controllers folder and select Add Controller
- You can elect to add “Create”, “Update”, and “Details” controller methods
- Action methods often return a view using the controller method `View()`.
 - If no arguments will return view of same name as method
 - You can supply names of both a view and a model to be used.

Adding New View

- Right-click on controller's action method name and select Add View
 - You can now elect to make the view strongly typed by selecting an existing model (the Add View dialog calls that "View Data Class")
 - You can elect to select View Content:
 - Create, Delete, Details, Edit, List
 - List, for example, provides a nicely structured table presentation of records from your model

Passing Data to Controller

- Controllers are created with each call
 - So you can't save data in controller method from call to call
 - You can save data in ViewData dictionary or in Session. Often you save all your data in a database.
 - When a controller method is called any model arguments are passed newly created instances of the named model.
 - On PostBack data from view is bound to model

Passing Data to a View

- There are two main ways of getting data from a controller method to its view:
 - Add data to ViewData dictionary, shared by controller and view
 - Pass model as argument of controller method View(aModel)

Returning Data from View

- Controller methods receive data in several ways:
 - A get request (from an ActionLink) may send an index into a model list.
 - A PostBack (post request) sends its form data back to the Asp.Net Request Dictionary.
 - PostBack data is automatically bound to the view model held by the method if the view is strongly typed.

That's All Folks!