Jim Fawcett
CSE686 – Internet Programming
Summer 2010

ASP.NET MVC

What is Asp.Net MVC?

- Framework for building web applications
- Based on Model-View-Controller pattern
  - Model manages the applications data and enforces constraints on that model.
    - Often accessed through persistent objects
  - Views are mostly passive presentations of application state.
    - Views generate requests sent to a controller based on client actions.
  - Controllers translate requests into actions on the data model and generate subsequent views.

MVC Life Cycle

- Clients request a named action on a specified controller, e.g.:
  - http://localhost/aController/anAction
- The request is routed to aController's anAction method.
  - That method decides how to handle the request, perhaps by accessing a model's state and returning some information in a view.

What is a Model?

- A model is a file of C# code and an associated data store, e.g., an SQL database or XML file.
  - The file of C# code manages all access to the application's data through objects.
  - Linq to SQL and Linq to XML create queries into these data stores
    - This can be direct
    - More often it is done through objects that wrap db tables or XML files and have one public property for each attribute column of the table.

FirstMVCDemoSu10 Model

```
namespace MvcApplication2.Models
{
  public class FileHandler
  {
    public string path { get; set; }
    public string[] files { get; set; }
    public bool GetFiles(string pattern)
    {
      try
      {
        int pos = path.LastIndexOf("Home");
        path = path.Substring(0, pos) + "Views\\Home";
        files = System.IO.Directory.GetFiles(path);
        return true;
      }
      catch { return false; }
    }
  }
}
```

What is a View?

- Views are usually aspx files with only HTML and inline code, e.g., <% … C# code here … %>.
  - Code is used just to support presentation and does no application processing.
  - The HTML is augmented by HTML Helpers, provided by Asp.Net MVC that provide shortcuts for commonly used HTML constructs.
  - Asp.Net MVC comes with jQuery (Javascript) libraries to support reacting to client actions and doing AJAX communication with the server.

FirstMVCDemoSu10 View

```
<%@ Page Language="C#" MasterPageFile="~/Views/Shared/Site.Master"
Inherits="System.Web.Mvc.ViewPage" %>

<asp:Content ID="aboutTitle" ContentPlaceHolderID="TitleContent" runat="server">
    About Us
</asp:Content>

<asp:Content ID="aboutContent" ContentPlaceHolderID="MainContent" runat="server">
    <h2>Demo Application</h2>
    <%
      try {
        string[] files = ((MvcApplication2.Models.FileHandler)Model).files;
        Response.Write("<br/>Find Files in Home Controller's Views Folder");
        foreach (string file in files)
          Response.Write("<br/>" + file);

        string path = ((MvcApplication2.Models.FileHandler)Model).path;
        System.IO.FileInfo fi = new System.IO.FileInfo(path + "\\Index.Aspx");
        Response.Write("<p/>Index.Aspx Last Revised: " + fi.LastAccessTime.ToString());
      }
      catch { Response.Write("<p/>Error finding path or file"); }
    %>
    <p>
        This application is intended to host a set of tutorial demos for Asp.Net MVC.
        Each Tab will open a new demo example.  Eventually I will segregate them into
        demo categories with a controller for each category.
    </p>
</asp:Content>
```

What is a Controller?

- A controller is a C# file with controller classes that derive from the class Controller.
  - A controller defines some category of processing for the application.
  - Its methods define the processing details.
  - Routing to a controller is defined in the Global.Asax.cs file. Its default processing is usually what you need.

FirstMVCDemoSu10 Controller

```
namespace MvcApplication2.Controllers
{
  [HandleError]
  public class HomeController : Controller
  {
    public ActionResult Index()
    {
      ViewData["Message"] = "First Model-View-Controller Demos";

      return View();
    }

    public ActionResult Form()
    {
      ViewData["Message"] = "Form not yet implemented";

      return View();
    }

    // code removed here to fit on slide

    public ActionResult About()
    {
      string path = Server.MapPath(".");
      Models.FileHandler fh = new Models.FileHandler();
      fh.path = path;
      fh.GetFiles("*.*");
      return View(fh);
    }
  }
}
```

Web Application Development

- Create a new Asp.Net MVC project
  - Delete any part of that you don't need
- Add a controller for each category of processing in your application:
  - A category is usually a few pages and db tables that focus on some particular application area
- Add methods to each controller for each request you wish to handle.
- Add views as needed for each controller action
- Add Model classes to support the application area:
  - Each model class has public properties that are synchronized with data in the model db or XML file.

An Opinion

- This Asp.Net MVC structure is very flexible:
  - You can have as many application categories as you need, simply by adding controllers.
  - The controllers keep the application well organized.
  - You can have as many views as you need. The navigation is simple and provided mostly by the MVC infrastructure, e.g., routing in Global.asax.cs.
  - You can have as many models as you need. Just add classes and use Linq to access the data.

Things you need to know

- LINQ – Language integrated query
  - Linq to XML and Linq to SQL are commonly used by models to provide data needed by a controller for one of its views.
- Jquery – Javascript Query library
  - Jquery is frequently used by views to react to client actions in the browser.
  - Jquery has facilities to use AJAX to retrieve small amounts of information from the server without loading a new view.

References

- Class Text: “Pro Asp.Net MVC”
- Asp.Net MVC Tutorials
  - http://weblogs.asp.net/scottgu/archive/2007/11/13/asp-net-mvc-framework-part-1.aspx
  - http://nerddinnerbook.s3.amazonaws.com/Intro.htm
- Linq:
  - http://dotnetslackers.com/articles/csharp/introducinglinq1.aspx
- Jquery
  - http://docs.jquery.com/Tutorials:How_jQuery_Works#jQuery:_The_Basics