



Module 3 Day 4

MVC Views 2

What makes an application?

- Program Data

- ✓ Variables & .NET Data Types
- ✓ Arrays
- ✓ More Collections (list, dictionary, stack, queue)
- ✓ Classes and objects (OOP)

- Program Logic

- ✓ Statements and expressions
- ✓ Conditional logic (if)
- ✓ Repeating logic (for, foreach, do, while)
- ✓ Methods (functions / procedures)
- ✓ Classes and objects (OOP)
- ❖ Frameworks (MVC)

- Input / Output

- User
 - ✓ Console read / write
 - ✓ HTML / CSS
 - ❑ Front-end frameworks (HTML / CSS / JavaScript)
- Storage
 - ✓ File I/O
 - ✓ Relational database
 - ❑ APIs

A little more on ASP Attributes

- **<a asp-area="" asp-controller="Home" asp-action="Index">Home**
 - Generates **Home**
- **<a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy**
 - Generates **Privacy**
- **<a asp-controller="Accounts" asp-action="Details" asp-route-id="5">Details**
 - Generates **Details**
- **<a asp-controller="Accounts" asp-action="List" asp-route-sort="Balance">Accounts**
 - Generates **Accounts**



Let's
Code

MVC Request Workflow

- Browser sends HTTP Request to server
- Framework *routes* request to a Controller Action
- Controller creates Model objects to do some work (fetch or update)
- Controller invokes the View
 - Passes Model data (possibly other properties) to the view
- View merges HTML + Model data to create response body
- Controller returns HTTP Response to the Browser
- Browser displays response body (in the case of HTML)

Passing Model Data - Controller

- Controller gets Model data, View needs it
- Controller passes Model using a View method overload
- Overloads of the View method:

`View()` : uses the default view (action name), no model data

`View("myView")` : uses view "myView", no model data

`View(myModel)` : uses default view, passes model myModel

`View("myView", myModel)` : uses view "myView", model myModel

Passing Model Data - View

- Add the **@model** directive to the top of the View file
@model SavingsAccount
 - Specifies the Type of the Model
 - Other Razor directives. Note all Razor directives are lower-case
 - @using
 - @namespace
 - @section
- Refer to the passed-in model object with the **Model** property
 - This is a property available in the Razor code
 - As a property it is declared as upper-case

Let's
Code

A little more on Layout Pages

- `_ViewStart.cshtml`
 - Code in this file runs before any view code runs
 - So every view does not need to set its Layout
- You can Render more content than just a single Body
 - `@RenderSection` is used in the Layout file

```
    </main>  
    @RenderSection("sidebar", false)  
  </body>
```

- `@section` is used in the View file to define blocks of code to be rendered

```
@section sidebar {  
    <aside>  
        Here is a side-bar.  
    </aside>  
}
```

Let's
Code