



Module 3 Day 5

MVC Views 3

Even More on Layout

- Partial Views

- A Razor file (*.cshtml*) that renders HTML output *within* another markup file's rendered output
- Used to place duplicate markup into multiple Views
 - E.g., a common form, or a product display tile
- Name starts with `_` by convention (not required)
- “Using” View includes the partial view with:
`<partial name=“_partialVwName” model=“Model” />`

- Example

- `_Filters` partial view
- Used in Index, Tile

Passing QueryString Parameters

- Asp-route-{variable-name} adds {variable-name} to the querystring on the URL
- When that URL is clicked, the framework does [Model Binding](#)
 - Maps URL parameters (query string) to Action parameters
 - Looks for names in the query string that match parameter name
 - But there's more magic: it also looks for public properties of Action parameter types, and matches those names if possible.
- Example:
 - Product sort order (int)
 - Product filtering (complex type)

ActionResult

- Actions are required to return *ActionResult*
- Controller View() method returns a *ViewResult*
- Other types of ActionResult:
 - RedirectResult, RedirectToActionResult
 - OkResult, NotFoundResult, CreatedResult, ForbidResult, StatusCodeResult
 - FileContentResult, FileStreamResult
 - JsonResult
 - And many more
- Examples
 - HomeController.Index returns a re-direct
 - ProductsController.Id should return a NotFound (404). Let's do that.

Dependency Injection

- Don't let this blow your mind!
- Inversion of Control pattern
- Removes the task of creating DAO's in the Controllers
- Tells the framework to create and pass DAO's into the controller
- Example:
 - ProductsController constructor