Module 3 Day 5

MVC Views 3 – Ends and Odds

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

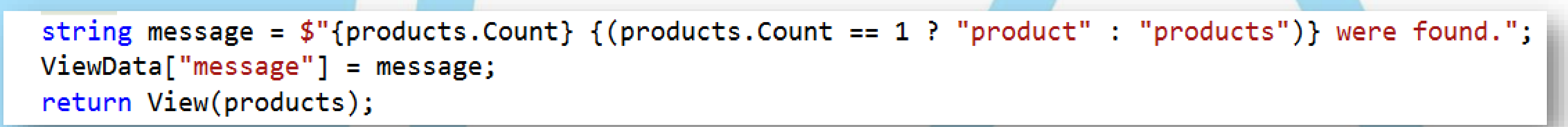
Passing QueryString Parameters

- 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:
  - Films Search

Let's Code

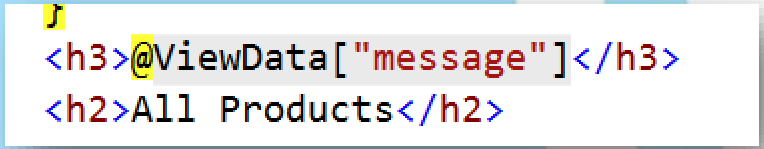
ViewData

- Another way to pass data from Controller to View
  - Not type-safe, no IntelliSense…
  - "Quick and dirty"
  - May be used for messages, for example
- In the Controller:



```
string message = $"{products.Count} {(products.Count == 1 ? "product" : "products")} were found.";
ViewData["message"] = message;
return View(products);
```

- In the View:



```
}
<h3>@ViewData["message"]</h3>
<h2>All Products</h2>
```

Let’s Code

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
  - _ListFilms partial view

Let’s Code

IActionResult

- Actions are required to return *IActionResult*
- Controller View() method returns a *ViewResult*
- Other types of IActionResult:
  - **RedirectResult**, RedirectToActionResult
  - OkResult, NotFoundResult, CreatedResult, ForbidResult, StatusCodeResult
  - FileContentResult, FileStreamResult
  - JsonResult
  - And many more

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:
  - FilmsController constructor

Let’s Code