

Module 3 Day 9

Form Validation

What makes an application?

- Program Data

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

- Program Logic

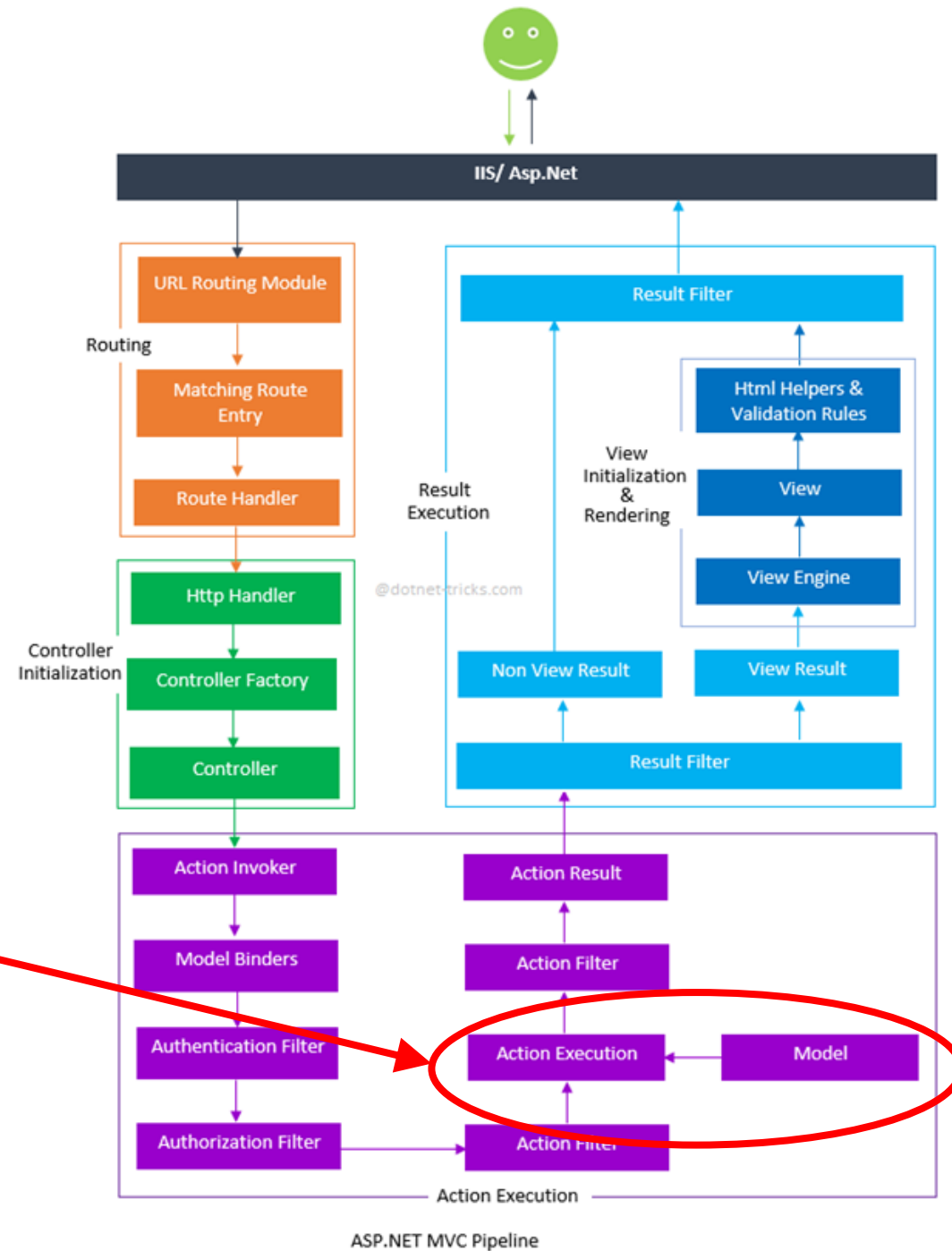
- ✓ Statements and expressions
- ✓ Conditional logic (if)
- ✓ Repeating logic (for, foreach, do, while)
- ✓ Methods (functions / procedures)
- ✓ Classes and objects (OOP)
- ❖ Frameworks (MVC)

- Input / Output

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

The "Pipeline"

You are here



TempData

- The love child of Session and ViewData
 - Accessed like ViewData, but
 - Stores data for this and next request (ViewData only lasts for **this** request)
- To Store: `TempData["myKey"] = "myValue";`
- To Retrieve: `string theData = (string)TempData["myKey"];`
- Often used in conjunction with Redirects to "pass" data
- Example: Add City



Let's
Code

Form Data Validation

- Client-side validation
 - Ensures data sent to server is good; if not, no data is sent
 - Tells the user there is an issue before data is sent to the server
 - Implemented using HTML5 controls and/or JavaScript
 - Provides a good user experience, **nice to have**
 - Can be bypassed
- Server-side validation
 - Implemented on server, after data is sent
 - Last layer of protection for the data, **must have**
 - Cannot be bypassed

Server-side Data Validation

- Model
 - Model defines the “rules” for valid data
 - [Attributes] are used to declare the rules
- The MVC Framework validates the data
 - Data is validated before the Controller is called
 - Validation can also be called when needed
- View
 - Accepts user data
 - Shows validation messages to the user
- Controller
 - Checks the status of validation and determines what to do
 - E.g., if validation errors exist, then re-display the view,
 - Else, update the database

Validation – The Model – Data Annotations

- [CreditCard]: Validates that the property has a credit card format
- [Compare]: Validates that two properties in a model match
- [EmailAddress]: Validates that the property has an email format
- [Phone]: Validates that the property has a telephone number format
- [Range]: Validates that the property value falls within a specified range
- [RegularExpression]: Validates that the property value matches a specified regular expression
- [Required]: Validates that the field is not null.
- [StringLength]: Validates that a string property value doesn't exceed a specified length limit
- [Url]: Validates that the property has a URL format
- [DataType(DataType.Date)] - DataType.EmailAddress, DataType.Password
- <https://docs.microsoft.com/en-us/dotnet/api/system.componentmodel.dataannotations?view=netcore-2.2>
- <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/validation?view=aspnetcore-2.2>



Let's
Code

Validation – The View – Tag Helpers

- `<input asp-for="Name" />`
 ``
- `<div asp-validation-summary="all"></div>`



Let's
Code

Validation – The Controller

- Framework validates and sets the Model State
- Controller must check Model State before acting
- If the Controller changes the model, `TryValidateModel(theModel)` can be called.

```
[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult New(Movie movie)
{
    // See if there are errors
    if (!ModelState.IsValid)
    {
        // Display the New View again
        // with the errors
        return View(movie);
    }

    // otherwise save the movie
    movieDal.Create(movie);
    return RedirectToAction("Success");
}
```

Let's
Code