Building Better Forms with Razor

Alex Wolf

www.alexwolfthoughts.com

To-Do List

Understanding Models in Razor

Demo: Building Forms with HTML Helpers

Model Binding in MVC

Demo: Stepping Through the Model Binding Process

Understanding Form Validation in MVC

Demo: Handling Client Side Validation

Demo: Handling Server Side Validation

Demo: Model Binding Collections

Working with Models in Razor

```
@model OrderingApplication.Models.Client
<div id="page-wrapper">
    <span>@Model.FirstName</span>
    <span>@Model.Email</span>
</div>
```

Defining the Model

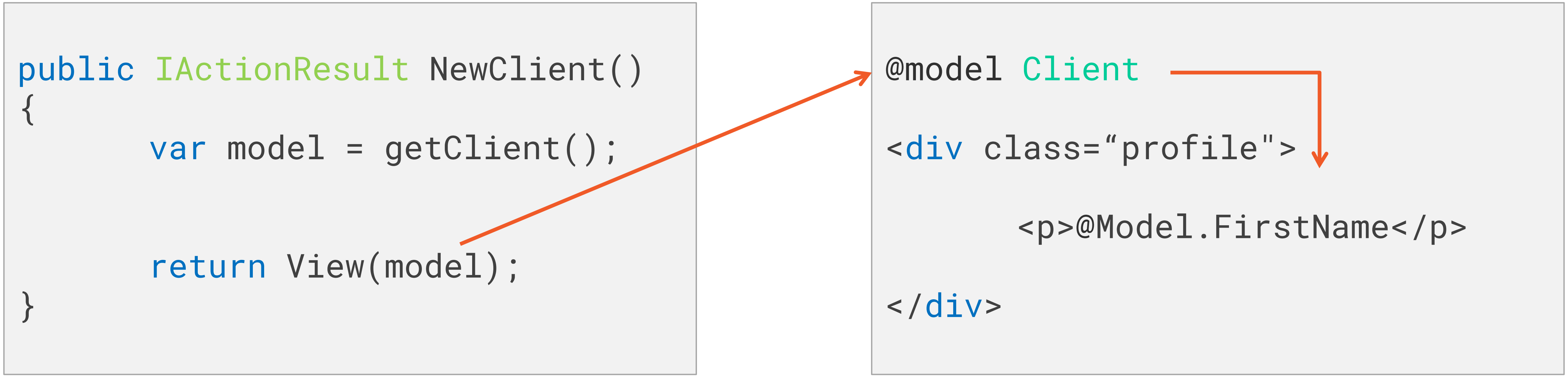
Declaring a Model allows its properties to be accessed inside the view

```
public class ClientController : Controller
{
    public IActionResult AllClients()
    {
        return View (“ClientList”)
    }

    public IActionResult NewClient()
    {
        // Retrieve from Database
        Client clientModel = GetClient();
        return View (clientModel)
    }
}
```

◀ Return a different, specific view

◀ Passes a Model to a view through the View helper method

Passing Data from an Action Method to a View



Client Controller

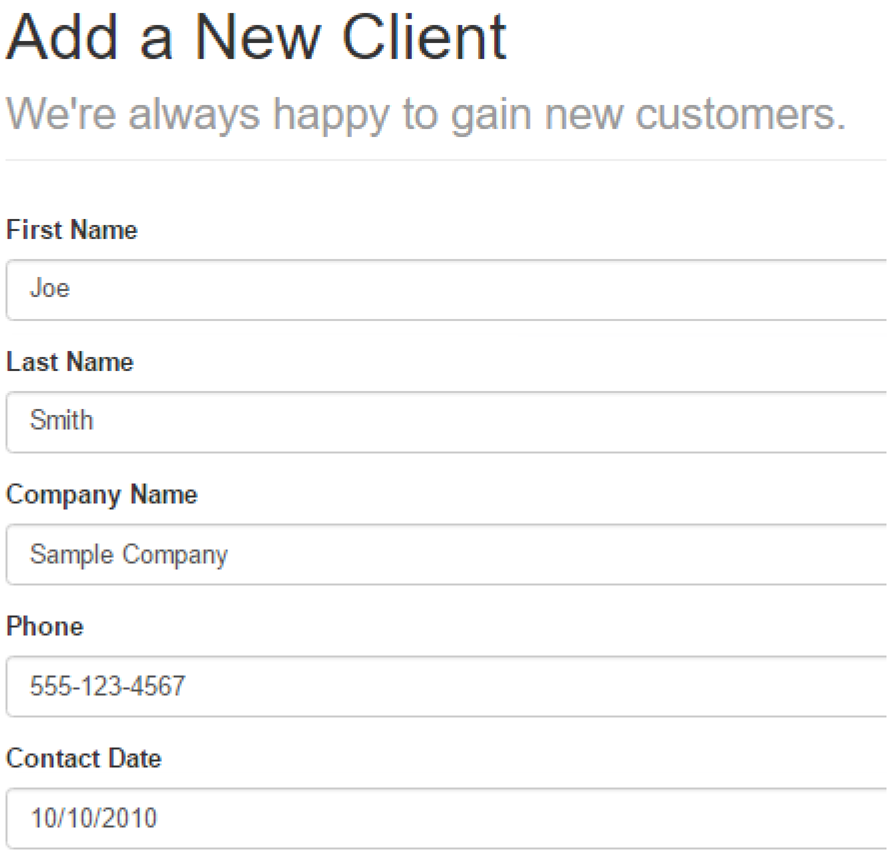
```
public IActionResult NewClient()
{
        var model = getClient();


        return View(model);
}
```

New Client View

```
@model Client

<div class=“profile">

      <p>@Model.FirstName</p>

</div>
```

Populating Form Fields in a View

Populated Model



Add a New Client

We're always happy to gain new customers.

First Name

Joe

Last Name

Smith

Company Name

Sample Company

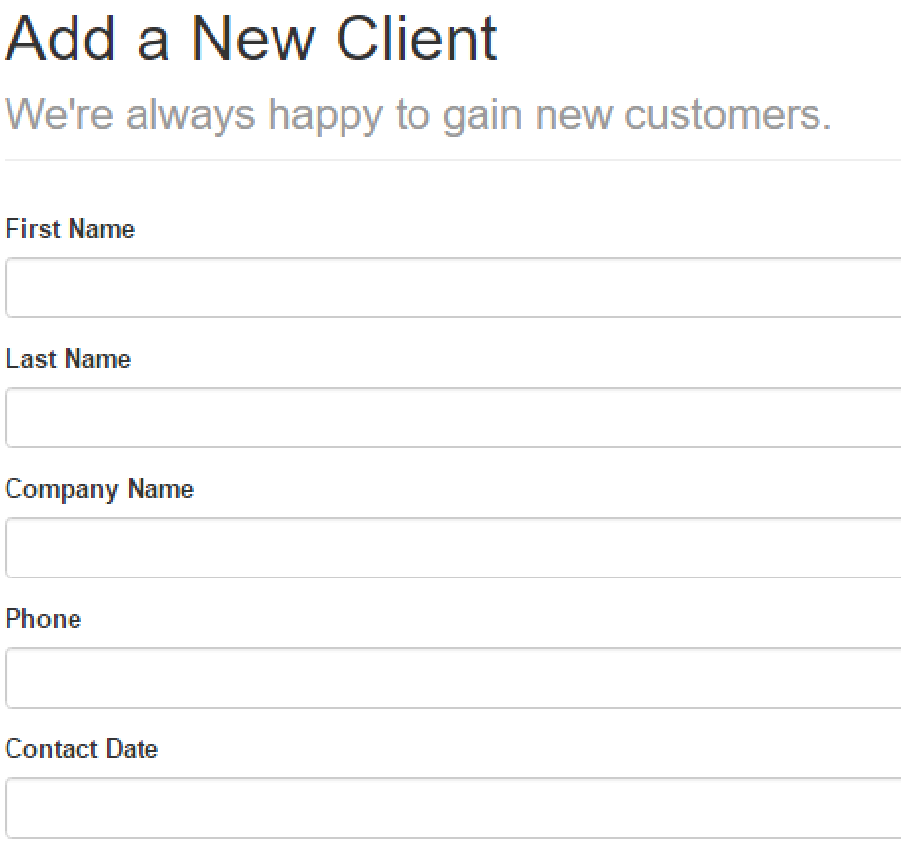
Phone

555-123-4567

Contact Date

10/10/2010

Unpopulated Model



Add a New Client

We're always happy to gain new customers.

First Name

Last Name

Company Name

Phone

Contact Date

Razor HTML Form Helpers

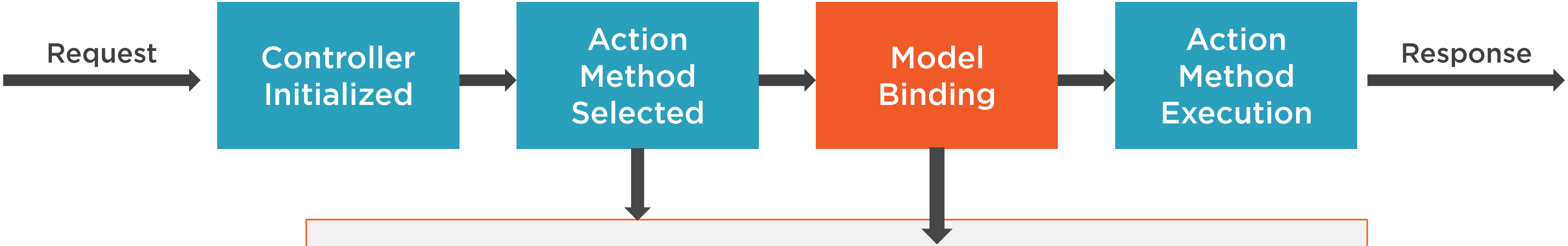
Strongly Typed HTML Form Helpers
@Html.TextBoxFor()
@Html.CheckBoxFor()
@Html.RadioButtonFor()
@Html.DropDownListFor()

Looking Ahead

Demo: Building Forms with HTML Helpers

Model Binding in MVC

The Model Binding Process



```
public IActionResult NewClient(Client newClient)
{
        return View();
}
```

The Default Value Providers in MVC

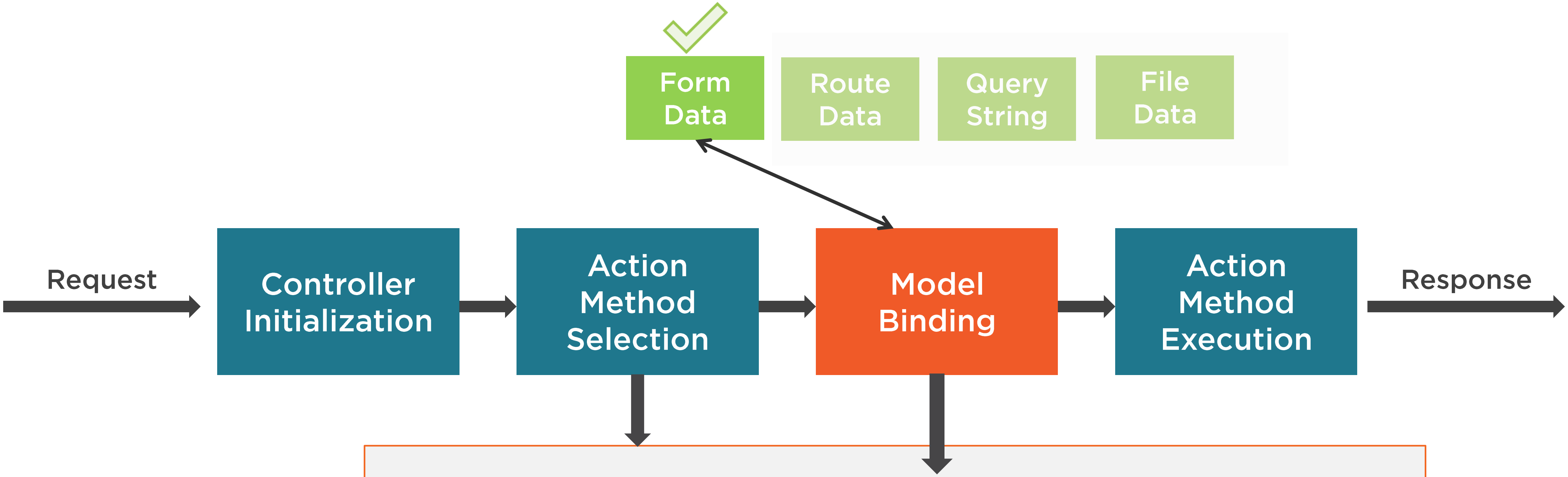
Form Value Provider

Route Data Value Provider

Query String Value Provider

File Value Provider

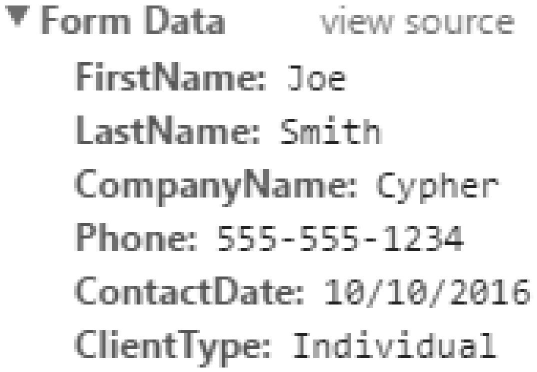
Supplying Data with Value Providers



```
public IActionResult NewClient(Client newClient)
{
        return View();
}
```

Binding Posted Form Data

HttpRequest (Simplified)



▼ Form Data   view source

FirstName: Joe

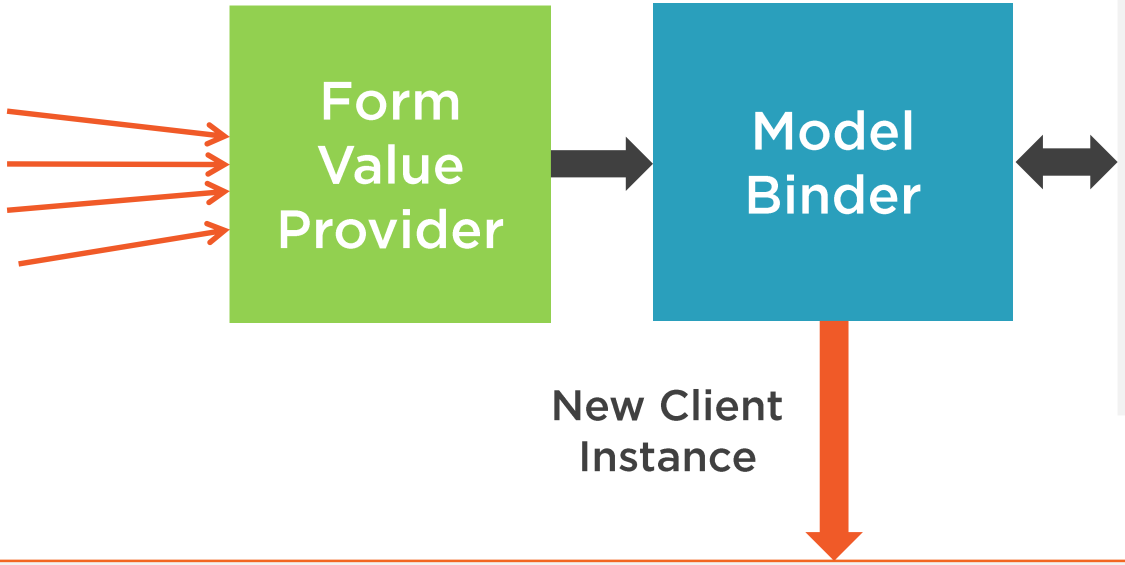
LastName: Smith

CompanyName: Cypher

Phone: 555-555-1234

ContactDate: 10/10/2016

ClientType: Individual



Client Class

```
public class Client{
  public string FirstName{}
  public string LastName{}
  public string CompanyName{}
  public string Phone{}
}
```

```
public IActionResult NewClient(Client newClient)
{
        return View();
}
```

```
<div id="page-wrapper">
    <input type=“text” name=“FirstName” class=“form-control” />
    <input type=“text” name=“BillingAddress.Zip” class=“form-control” />
</div>
```

Naming Elements for Model Binding

Razor helps ensure form elements are named for easy Model Binding

Demo: Model Binding with Razor Forms

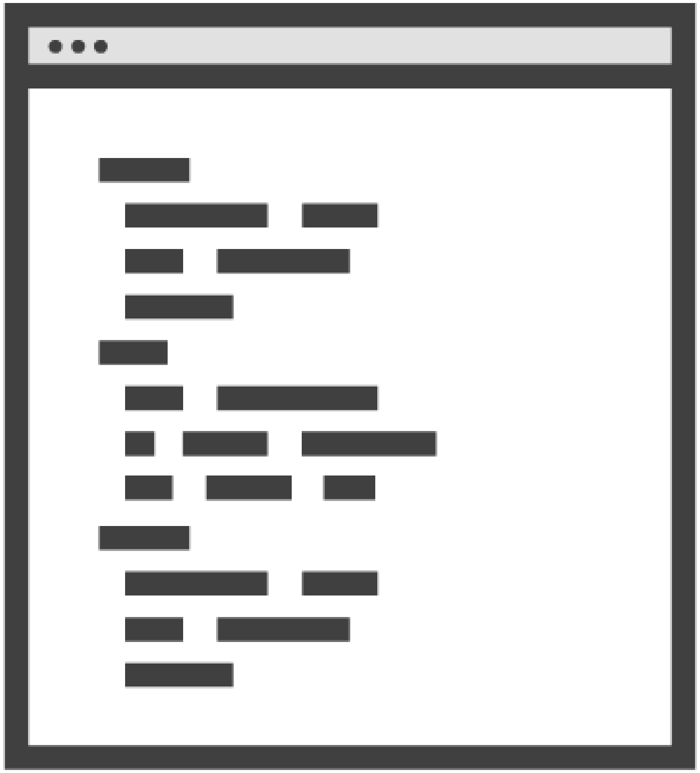
Form Validation with Razor

Client and Server Validation

Client Validation

User Experience

Server Validation



Data Security

```
public class Client
{
    [Required]
    public string FirstName { get; set; }

    [Required]
    public string LastName { get; set; }

    [EmailAddress]
    public string EmailAddress { get; set; }

    [MaxLength(25)]
    public string Notes { get; set; }
}
```

Some type of value is required

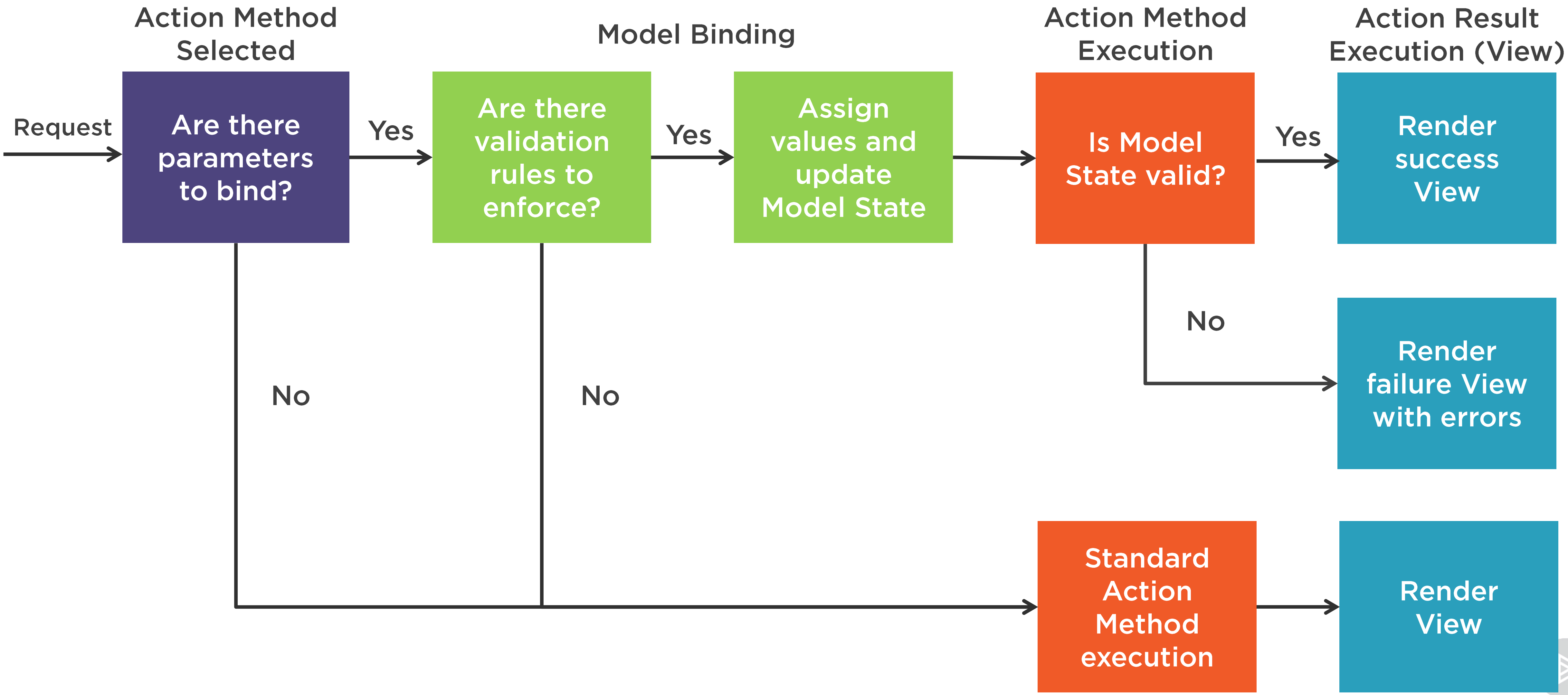
Value requires specific
formatting or length

```
<div id="page-wrapper">

    <input type=“text” data-val-required=“Field is Required”
    name=“FirstName” class=“form-control” />

    <input type=“text” data-val-email=“Field is not a valid email”
    name=“Email” class=“form-control” />

</div>
```

Client Validation with HTML Form Helpers

Razor HTML Helpers use Model meta data to render validation attributes on elements

Server Side Form Validation


Action Method Selected
Model Binding
Action Method Execution
Action Result Execution (View)
Request
Are there parameters to bind?
Yes
Are there validation rules to enforce?
Yes
Assign values and update Model State
Is Model State valid?
Yes
Render success View
No
Render failure View with errors
No
No
Standard Action Method execution
Render View

```
<div class="form-group">
    @Html.LabelFor(x => x.CompanyName, “Company Name")
    @Html.TextAreaFor(x => x.CompanyName)
    @Html.ValidationMessageFor(x => x.CompanyName)
</div>
```

Razor Validation Helpers

ValidationMessageFor renders property level errors

```
@Html.ValidationSummary()

<div class="form-group">

    @Html.LabelFor(x => x.CompanyName, “CompanyName")

    @Html.TextAreaFor(x => x.CompanyName)

</div>
```

Razor Validation Helpers (cont.)

ValidationSummary renders Model and (optionally) Property level errors

Demo: Client Side Validation with Razor

Demo: Server Side Validation with Razor

Demo: Remote Validation with Razor

Demo: Binding to Collections

Summary

Forms in Razor can be backed by a Model for strong typing and accessing data

HTML Helpers ensure form elements are named correctly for Model Binding

Razor Helpers integrate with Model Meta Data and jQuery libraries to render validation messages

Razor can bind to complex types and collections using the right syntax