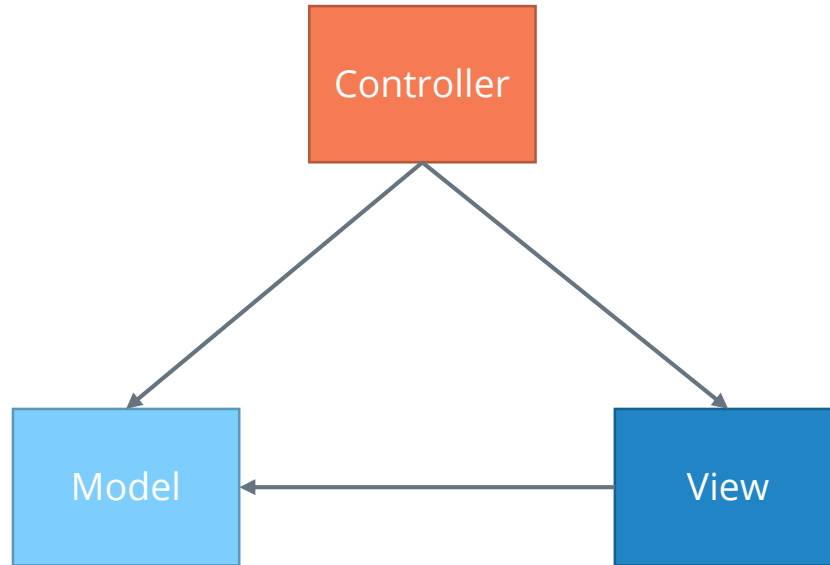




Headspring

Introduction to Views

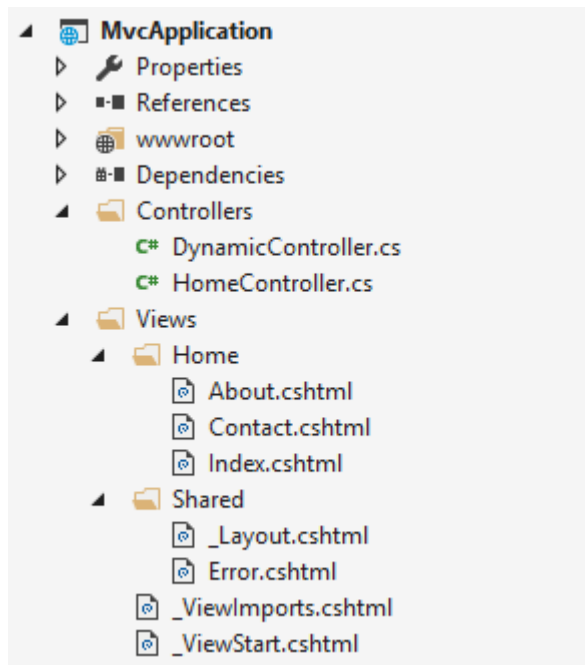
| What is a view?



| View Responsibilities

- Render the view model
- The end

MVC Solution Layout



View

```
@model ContactList.Features.Contacts.ContactViewModel[]
```



```
@{
```

```
    ViewBag.Title = "Contacts";
```

```
}
```



```
@Html.Button("+", Url.Action("Add"))
```

```
@foreach (var contact in Model)
```

```
{
```



```
<div class="contact-card"
```

```
    data-url="@Url.Action("Edit", new{ contact.Id})">
```

```
    @Html.Partial("ContactCard", contact)
```

```
</div>
```

```
}
```

```
<a href="/Contacts/Add" role="button">+</a>

<div class="contact-card" data-url="/Contacts/Edit/2a1d924e-6
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title">Johnson, Abe</h3>
    </div>
    <div class="panel-body">
      <div>
        <label for="Email">Email</label>:
        abe.johnson@example.com
      </div>
      <div>
        <label for="PhoneNumber">PhoneNumber</label>:
        555-123-0004
      </div>
    </div>
  </div>
```

```
public IActionResult Create()  
{  
    return View();  
}
```

```
@model ContactList.Features.Contacts.AddContactForm
@{
    ViewBag.Title = "Add Contact";
}
<h2>Add Contact</h2>
@using (Html.BeginForm())
{
    @Html.ValidationSummary()

    @Html.Input(x => x.FirstName)
    @Html.Input(x => x.LastName)
    @Html.Input(x => x.Email)
    @Html.Input(x => x.PhoneNumber)

    <div>
        @Html.SaveButton()
        @Html.CancelButton(Url.Action("Index"))
    </div>
}
```



```
<h2>Add Contact</h2>
▼ <form action="/Contacts/Add" method="post">
  ► <div class="validation-summary-valid" data-valmsg-summary="true">...</div>
  ▼ <div class="form-group">
    <label for="FirstName">First Name</label>
    <input class="text-box single-line form-control" data-val="true" data-val-
      required="The First Name field is required." id="FirstName" name=
        "FirstName" type="text" value>
  </div>
  ► <div class="form-group">...</div>
  ► <div class="form-group">...</div>
  ► <div class="form-group">...</div>
  ▼ <div>
    <button type="submit" class="btn btn-primary">Save</button>
    <a href="/Contacts" role="button" class="btn btn-default">Cancel</a>
  </div>
</form>
```

Review

- Incoming requests become action method calls in a controller.
- Inputs from the browser are bound to the method's inputs by naming convention.
- The returned object describes the HTTP response: render a view, redirect, etc.
- View files are located by action method name.

Built-in HTML Shorthand

```
@model ContactList.Features.Contacts.EditContactForm

@Html.DisplayFor(x => x.FirstName) //Display the value.

@using (Html.BeginForm())
{
    @Html.ValidationSummary() //Show error messages.

    @Html.HiddenFor(x => x.Id) //Hidden input field.

    @Html.LabelFor(x => x.FirstName) //User-facing label.
    @Html.EditorFor(x => x.FirstName) //Textbox, etc.

    //etc...

    <button type="submit">Save</button>
}
```

| Keep Views Dumb!

- Views are the hardest thing to test, and the easiest thing to get out of hand.
- Minimize conditional logic in a View.
- Minimize nested looping in a View.
- Minimize repetition in a View.
- Be RUTHLESS about keeping Views simple!

Views in MVC Core

| _ViewStart.cshtml

@{

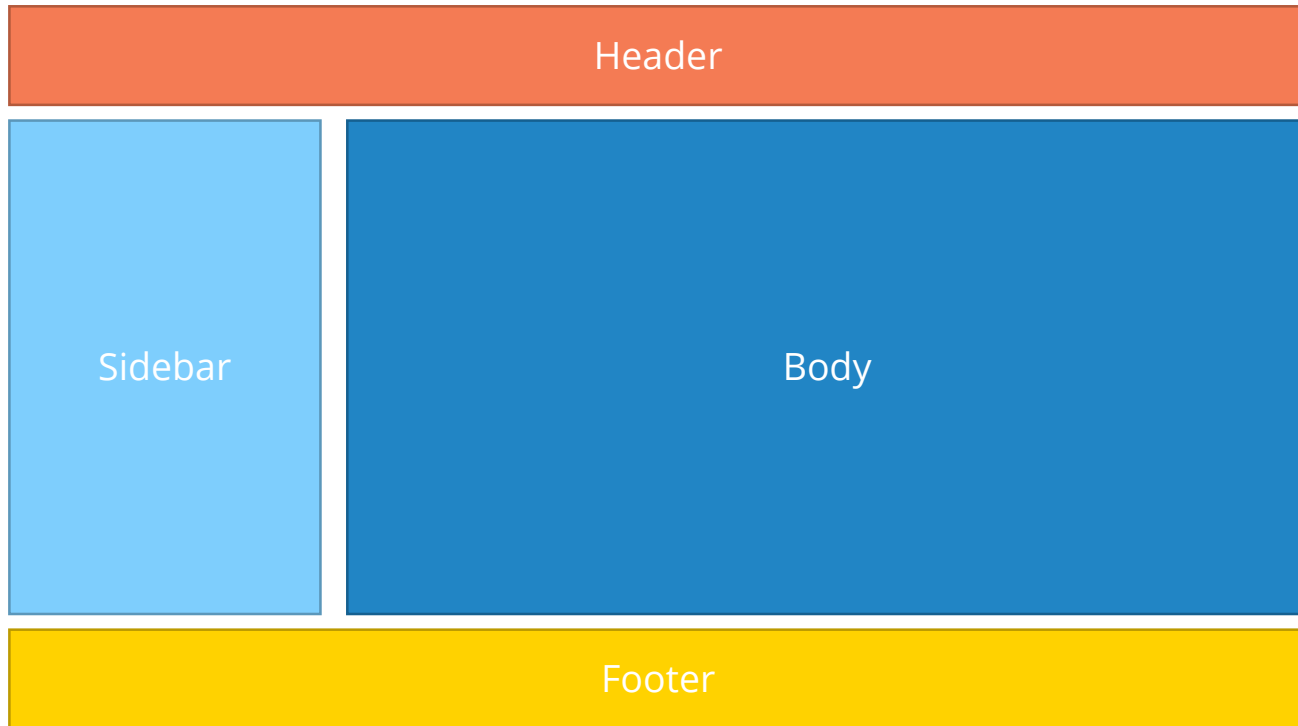
```
Layout = "_Layout";
```

}

| _ViewImports.cshtml

```
@using MvcApplication
@using MvcApplication.Models
@using MvcApplication.Models.AccountViewModels
@using MvcApplication.Models.ManageViewModels
@using Microsoft.AspNetCore.Identity
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```

| Layouts



| Layout body

```
<div class="container body-content">  
  @RenderBody()  
  <hr />  
  <footer>  
    <p>&copy; 2016 - MvcApplication</p>  
  </footer>  
</div>
```

| Layout sections

```
@RenderSection("scripts", required: false)  
</body>  
-  
</html>  
-
```

| Layout sections

```
@section Scripts {  
    <script>  
        window.alert("Hello world!");  
    </script>  
}
```

Tag Helpers

| What are tag helpers?

- A new way to build HTML snippets
- Supplement HTML Helpers
- HTML-friendly development experience
- Better IntelliSense than HTML Helpers
- I'm....not a fan

| HTML Helper with expressions

```
@Html.LabelFor(m => m.Email,  
    new { @class = "col-md-2 control-label" })
```

| Tag Helper

```
<label asp-for="Email"  
        class="col-md-2 control-label">  
</label>
```

| HTML Helper input

```
@Html.EditorFor(m => m.Password,  
    new { @class = "form-control" })
```


| Tag Helper form input

```
<input asp-for="Password"  
class="form-control"/>
```

| HTML Helper form

```
@using (Html.BeginForm("Register", "Account",  
    null, FormMethod.Post))  
{
```

| Tag Helper form

```
<form asp-controller="Account"  
      asp-action="Register"  
      asp-route-returnurl="@ViewData["ReturnUrl"]"  
      method="post" class="form-horizontal">
```

| Tag Helper prefixes

@tagHelperPrefix "asp:"

| Tag Helper prefixes

```
<asp:form asp-controller="Account"  
  asp-action="Register"  
  asp-route-returnurl="@ViewData["ReturnUr1"]"  
  method="post" class="form-horizontal">
```

```

@using (Html.BeginForm("Register", "Account", null,
    FormMethod.Post, null, new { @class = "form-horizontal" }))
{
    <h4>Create a new account.</h4>
    <hr/>
    @Html.ValidationSummary(null, new { @class="text-danger" })
    <div class="form-group">
        @Html.LabelFor(m => m.Email, new { @class= "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.EditorFor(m => m.Email, new { @class="form-control" })
            @Html.ValidationMessageFor(m => m.Email, null, new { @class="text-danger" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(m => m.Password, new { @class= "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.EditorFor(m => m.Password, new { @class="form-control" })
            @Html.ValidationMessageFor(m => m.Password, null, new { @class="text-danger" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(m => m.ConfirmPassword, new { @class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.EditorFor(m => m.ConfirmPassword, new { @class = "form-control" })
            @Html.ValidationMessageFor(m => m.ConfirmPassword, null, new { @class = "text-danger" })
        </div>
    </div>
}

```

```

<form asp-controller="Account" asp-action="Register"
      asp-route-returnurl="@ViewData["ReturnUrl"]"
      method="post" class="form-horizontal">
  <h4>Create a new account.</h4>
  <hr />
  <div asp-validation-summary="All" class="text-danger"></div>
  <div class="form-group">
    <label asp-for="Email" class="col-md-2 control-label"></label>
    <div class="col-md-10">
      <input asp-for="Email" class="form-control" />
      <span asp-validation-for="Email" class="text-danger"></span>
    </div>
  </div>
  <div class="form-group">
    <label asp-for="Password" class="col-md-2 control-label"></label>
    <div class="col-md-10">
      <input asp-for="Password" class="form-control" />
      <span asp-validation-for="Password" class="text-danger"></span>
    </div>
  </div>
  <div class="form-group">
    <label asp-for="ConfirmPassword" class="col-md-2 control-label"></label>
    <div class="col-md-10">
      <input asp-for="ConfirmPassword" class="form-control" />
      <span asp-validation-for="ConfirmPassword" class="text-danger"></span>
    </div>
  </div>
</div>

```

| Built-in HTML Helpers

- Form helpers
- Validation helpers
- Link helpers
- Security helpers

| Built-in Tag Helpers

- Form
- Input
- Textarea
- Label
- Validation
- Select