Partial views

Часто возникает необходимость в использовании одних и тех же фрагментов дескрипторов Razor и разметки HTML в разных местах приложения. Вместо дублирования содержимого можно применять **частичные представления**.

Их можно и нужно встраивать в другие обычные представления. Особенно удобно использовать частичные представления для отображения результатов AJAX-запросов, так как они динамически обновляют часть страницы без перезагрузки.

По действию похоже на обычное представление, но для него по умолчанию не определяется мастер-страница.

Способы внедрения:

За рендеринг частичных представлений отвечает объект PartialViewResult, который возвращается **методом PartialView().**

Чтобы встроить частичное представление в какое-то другое используется **хелпер HTML.Partial**.

@Html.Partial("Partial")

Html.RenderPartial - хелпер также принимает имя представления.

@{Html.RenderPartial("Partial");}

Он используется не в строчных выражениях кода Razor, а в блоке кода - то есть обрамляется фигурными скобками. Еще одно отличие между двумя способами заключается в том, что Html.RenderPartial напрямую пишет вывод в выходной поток, поэтому может работать чуть быстрее, чем Html.Partial.

Также как и в случае с обычными представлениями, мы можем создавать строго типизированные частичные представления, указав в шапке файла директиву @model.

Хелперы

HTML Helpers — это классы, которые помогают визуализировать HTML. Эти классы имеют методы, которые генерируют HTML во время выполнения. Мы также можем привязать объект модели к отдельному элементу HTML для отображения или извлечения значений.

Одно из основных различий между вызовом методов HtmlHelper и использованием тега HTML заключается в том, что методы HtmlHelper разработаны для упрощения привязки к данным представления или данным модели.

@Html используется для доступа к помощнику HTML, однако HTML является свойством HtmlHelpers, которое включено в базовый класс.

Виды: строчные, с#, tag.

ПОДРОБНЕЕ

**Частичные представления**

**Что такое частичные представления?**

Частичные представления — это отдельные файлы представлений, которые содержат фрагменты HTML и Razor-кода, которые могут быть переиспользованы в разных местах приложения. Они позволяют избежать дублирования кода и упрощают поддержку и тестирование.

**Как создавать частичные представления**

1. **Создание файла частичного представления**:
   * В папке Views вашего контроллера (например, Views/Calc) создайте новый файл с именем, начинающимся с символа \_. Например, \_CalcLink.cshtml.
2. **Содержимое частичного представления**:
   * Внутри файла определите HTML-код и Razor-синтаксис, которые хотите использовать. Например:

html

Copy

<p>

<a href="@Url.Action("Index", "Calc")">Index</a>

<a href="@Url.Action("Sum", "Calc")">Sum</a>

<a href="@Url.Action("Sub", "Calc")">Sub</a>

<a href="@Url.Action("Mul", "Calc")">Mul</a>

<a href="@Url.Action("Div", "Calc")">Div</a>

</p>

**Как подключать частичные представления**

* Чтобы использовать частичное представление в другом представлении, используйте хелпер Html.Partial или Html.PartialAsync:

csharp

Copy

@await Html.PartialAsync("\_CalcLink")

* Это вставит содержимое частичного представления \_CalcLink.cshtml в текущее представление.

**Виды частичных представлений**

1. **Статические частичные представления**:
   * Содержат фиксированный HTML-код и Razor-синтаксис.
2. **Динамические частичные представления**:
   * Принимают параметры и могут изменять отображаемое содержимое в зависимости от переданных данных.

**Строготипизированные представления**

**Что такое строготипизированные представления?**

Строготипизированные представления — это представления, которые используют модели для передачи данных. Вместо использования ViewBag для передачи данных в представление, вы определяете модель и передаете её в представление.

**Как создавать строготипизированные представления**

1. **Создание модели**:
   * Создайте класс модели, который будет представлять данные. Например:

csharp

Copy

public class CalculatorModel

{

public float? X { get; set; }

public float? Y { get; set; }

public float? Z { get; set; }

public string Press { get; set; }

public string Error { get; set; }

}

1. **Создание представления**:
   * Определите представление, используя модель:

csharp

Copy

@model CalculatorModel

1. **Использование модели в представлении**:
   * Теперь вы можете использовать свойства модели для отображения и ввода данных:

html

Copy

<input type="text" name="x" value="@Model.X" />

<input type="text" name="y" value="@Model.Y" />

<input type="text" readonly="readonly" value="@Model.Z" />

**Преимущества строготипизированных представлений**

* **Безопасность типов**: Ошибки типов обнаруживаются на этапе компиляции.
* **Читаемость**: Код более понятен, так как можно видеть, какие данные ожидаются в представлении.
* **Поддержка IntelliSense**: В средах разработки, таких как Visual Studio, предоставляется поддержка автодополнения для свойств модели.

**Хелперы**

**Что такое хелперы?**

Хелперы — это методы, которые упрощают создание HTML-кода в представлениях. Они позволяют сделать код более читаемым и поддерживаемым. В ASP.NET MVC есть два основных типа хелперов:

1. **HTML-хелперы**:
   * Предоставляют методы для генерации HTML-кода. Например, Html.TextBoxFor, Html.LabelFor, Html.CheckBox, и т.д.
   * Пример использования:

csharp

Copy

@Html.TextBoxFor(model => model.X)

1. **Собственные хелперы**:
   * Вы можете создавать свои собственные хелперы, чтобы инкапсулировать повторяющийся код.
   * Пример создания собственного хелпера:

csharp

Copy

public static class HtmlHelpers

{

public static IHtmlContent CustomButton(this IHtmlHelper htmlHelper, string text, string cssClass)

{

var tagBuilder = new TagBuilder("button");

tagBuilder.InnerHtml.Append(text);

tagBuilder.AddCssClass(cssClass);

return new HtmlString(tagBuilder.RenderStartTag() + tagBuilder.InnerHtml + tagBuilder.RenderEndTag());

}

}

**Как использовать хелперы?**

* Используйте HTML-хелперы в представлениях для создания элементов форм и других HTML-элементов, что делает ваш код более чистым и читаемым. Например:

csharp

Copy

@Html.LabelFor(model => model.X)

@Html.TextBoxFor(model => model.X)

**Заключение**

Частичные представления, строготипизированные представления и хелперы — это важные аспекты разработки на ASP.NET MVC, которые помогают организовать и структурировать код, улучшить его читаемость и поддерживаемость. Их правильное использование позволяет создавать более гибкие и устойчивые приложения.

Виды тег, html, валидации, кастомные(свои), форма

Эти ссылки используют хелпер Url.Action, чтобы создать URL-адреса, которые соответствуют действиям контроллера Calc. Это более предпочтительный подход, так как он автоматически генерирует правильные URL для заданных действий. Эти ссылки также не являются частичным представлением, а представляют собой просто HTML-код с динамически сгенерированными URL.

Таким образом, pattern объединяет в себе правила маршрутизации, которые позволяют определять, как HTTP-запросы сопоставляются с контроллерами и действиями в приложении ASP.NET Core. Это позволяет создавать понятные, читаемые и легко управляемые URL для вашего приложения.