**Tag Helper, Razor-Syntax & ViewData**

**Razor-Syntax**

**Razor** ist die serverseitige **Template-Sprache** in ASP.NET Core. Sie verbindet HTML mit C#-Code.

Razor-Dateien enden auf **.cshtml**

C#-Code wird mit **@** eingebettet

**Html Helper**

**Beispiel:**

**@**using (Html.BeginForm("Erfassen", "Buch", FormMethod.Post))

{

<label for="Titel">**@**Html.DisplayNameFor(model => model.Titel)</label>

**@**Html.TextBoxFor(model => model.Titel)

<button type="submit">Speichern</button>

}

**Tag Helper**

**Tag Helper** sind HTML-Attribute mit serverseitiger Logik. Sie verbessern Lesbarkeit, Struktur und IntelliSense.

**Beispiel:**

<form **asp-action**="Erfassen" **asp-controller**="Buch" method="post">

<label **asp-for**="Titel"></label>

<input **asp-for**="Titel" />

<button type="submit">Speichern</button>

</form>

**Vergleich: Razor vs. Tag Helper**

|  |  |  |
| --- | --- | --- |
| Eigenschaft | Razor-Syntax | Tag Helper |
| Struktur | Textbasiert mit @ | HTML-Tag mit speziellen Attributen |
| Lesbarkeit | Weniger HTML-artig | HTML bleibt HTML |
| IntelliSense | Eingeschränkt | Vollständig |
| Modellbindung | @Html.TextBoxFor(...) | <input asp-for="..." /> |

**Vergleich: Razor vs. Tag Helper im Bezug auf das Bücherverwaltung-Projekt**

|  |  |  |
| --- | --- | --- |
| Funktion | Tag-Helper-Syntax | Html-Helfer-Syntax (klassisch) |
| Formular öffnen | <form asp-action="Erfassen" asp-controller="Buch" method="post"> | @using (Html.BeginForm("Erfassen", "Buch", FormMethod.Post)) { |
| Label | <label asp-for="Titel"></label> | @Html.LabelFor(model => model.Titel) |
| Textfeld | <input asp-for="Titel" /> | @Html.TextBoxFor(model => model.Titel) |
| Zahleneingabe | <input asp-for="Erscheinungsjahr" type="number" /> | @Html.TextBoxFor(model => model.Erscheinungsjahr, new { type = "number" }) |
| Validierungsnachricht | <span asp-validation-for="Titel"></span> | @Html.ValidationMessageFor(model => model.Titel) |
| Button | <button type="submit">Speichern</button> | <button type="submit">Speichern</button> (gleich) |

**Was geht – und was nicht**

|  |  |  |
| --- | --- | --- |
| Aufgabe | Möglich mit @foreach | Möglich mit Tag-Helper |
| Liste durchlaufen | Ja | Nein |
| HTML-Tags dynamisch erzeugen | Ja | Nein direkt |
| HTML-Tags anreichern (z. B. Links, Formulare) | Ja | Ja (Tag-Helper innerhalb der Schleife) |

**Hinweis**

Tag Helper werden erst aktiviert, wenn @addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers im \_ViewImports.cshtml steht.

ViewData ist nicht dasselbe wie TempData oder Session – es lebt **nur für einen Request**.

Razor ist die Basis – Tag Helper bauen darauf auf.

**ViewData**

**Definition:** ViewData ist ein Dictionary (string -> object), das Daten vom Controller an die View überträgt.

**Controller:**

ViewData["Seitentitel"] = "Neues Buch erfassen";

**View:**

<h2>@ViewData["Seitentitel"]</h2>

**Eigenschaften:**

Funktioniert innerhalb **eines Requests**

Wird **nicht** zwischen Requests gespeichert

Kein typisiertes Objekt (im Gegensatz zum Model)

Geeignet für einfache Informationen

**Intern:**

Typ: ViewDataDictionary

Lebt im ViewContext des aktuellen Requests

Wird bei return View() an die Razor View übergeben

Wird nach der Antwort an den Client verworfen

|  |  |
| --- | --- |
| Thema | Zusammengefasst |
| Razor | C# in HTML mit @, flexibel für Logik |
| Tag Helper | HTML-Attribute mit Razor-Logik, moderner |
| ViewData | Dictionary zur Datenübergabe Controller → View |

**Übung**

Erstellen Sie ein Formular zur Erfassung eines Films mit den Feldern:

* Titel (Text)
* Regisseur (Text)
* Erscheinungsjahr (Zahl)

Nutzen Sie:

* Tag Helper für das Formular
* ViewData für den Seitentitel
* Razor für eine Bedingung: Wenn keine Filme vorhanden sind, zeigen Sie "Kein Film gefunden."