

ASP.NET Core MVC

Razor

Ingo Köster

Diplom Informatiker (FH)

Was ist Razor?

- › Auszeichnungssprache, die serverbasierten Code (z.B. in C #) in Web-Seiten (HTML) einbetten kann
- › Server-Code kann dynamische Web-Inhalte zur Laufzeit erstellen
- › Wird eine Web-Seite aufgerufen, führt der Server den Server-basierten Code in der Seite aus, bevor er die Seite zurück an den Client-Browser sendet

Razor

› Aus

```
<ul>  
  @foreach (var p in products) {  
    <li>@p.Name</li> }  
</ul>
```

› Wird

```
<ul>  
  <li>Phaser</li>  
  <li>Lightsaber</li>  
  <li>Pulse Rifle</li>  
</ul>
```

Razor-Syntax

- › Server-seitiger Code wird immer durch ein @-Symbol eingeleitet
- › Verwendung:
 - › @ zum Markieren von serverseitigem C#-Code
 - › @@ um das Zeichen @ in HTML darzustellen
 - › @: um explizit einen Textabschnitt zu markieren
 - › Beispiel
 - › @if (DateTime.Now.Year > 2010) {
 - › @: Text ohne HTML-Elemente (nur einzeilig) }
 - › @* Ein Kommentar in Razor (auch über viele Zeilen) *@

Razor-Syntax - <text> Element

- › Mittels des Elements <text> können längere Textpassagen angezeigt werden

<text>

Ganz viel Text, welcher auch über mehrere Zeilen gehen kann und dabei trotzdem das aktuelle Datum enthalten kann:

@DateTime.Now

</text>

- › Um Texte ohne HTML-Codierung auszugeben, kann die Hilfsmethode `Html.Raw()` verwendet werden

Razor - Block

› Ein Block in Razor beginnt mit einem @-Symbol

› Beispiel

```
@( "Heute: " + DateTime.Now.ToString() )
```

› Der Inhalt eines einfachen Razor-Blocks „@()“ wird gerendert und sollte daher einen String enthalten bzw. in einen String überführt werden können

Razor – Multi-Line Block (Code Block)

- › Ein Multi-Line Block beginnt mit dem @-Symbol, gefolgt von geschweiften Klammern
- › Dieser kann Anweisungen enthalten, diese werden nicht gerendert

```
@{  
    int number = 1;  
    string message = "Number is " + number;  
}
```

```
<p>Your Message: @message</p>
```

Kommentare

- › Innerhalb von Razor Blöcken können C# Kommentare verwendet werden

```
@{  
    /* C# Kommentar */  
    // C# Kommentar  
}
```

- › So wie die Razor Kommentare @* *@ werden diese Inhalte nicht in das Ausgabe-HTML eingefügt

Razor – Verzweigungen

› @if leitet eine Verzweigung ein

```
@if (products.Count == 0)
{
    <p>Sorry – no products in this category</p>
}
else
{
    <p>We have a product for you!</p>
}
```

Razor – Verzweigungen (Fortsetzung)

```
@switch (zahl)
{
    case 1:
        <p>Wert der Zahl ist 1!</p>
        break;
    case 42:
        <p>Wert der Zahl ist 42!</p>
        break;
    default:
        <p>Die Zahl hat irgendeinen Wert.</p>
        break;
}
```

Razor – Schleifen

- › Es gibt die bekannten Schleifen @foreach und @for, @while und @do while

```
<ul id="products">  
    @foreach (var p in products)  
    {  
        <li>@p.Name (@p.Price €)</li>  
    }  
</ul>
```

Razor – Schleifen (Fortsetzung)

```
@{ var i = 0; }
```

```
@do  
{  
    i++;  
} while (i < 10);
```

```
@{ var i = 0; }
```

```
@while (i < 10)  
{  
    i++;  
}
```

Razor – using

- › Mittels `@using` kann ein Namensraum in eine View eingebunden werden
- › Innerhalb von Razor Anweisungen steht dieser Namensraum anschließend zur Verfügung

```
@using MVC_Projekt.Models
```

```
...
```

```
@{
```

```
    MeinModel m = ...
```

```
}
```

Razor - Funktionen

- › Es ist möglich in Razor Funktionen anzulegen (mit @functions)
- › Auf dieses Mittel sollte in einer View im allgemeinen verzichtet werden!!!
 - › Eine View sollte keine Logik enthalten
- › Für Helper und Vorlagen können Funktionen jedoch nützlich sein

```
@functions
{
    private string Hallo(string wert)
    { return "Hallo " + wert; }
}
```

```
<b>Der Text:</b> @Hallo("Welt")
```