ASP MVC Html Template Erstellen und verwenden

Dazu erstellt man in den Ordner Views eine neue cshtml Datei, welche als Template dienen soll. Um diese vor anderen Views zu trennen empfiehlt es sich diese Template cshtml Seiten im shared Verzeichnis zu sichern.

Dieses Template kann z.B. ein Logo und den Footer beinhalten, die jede oder nur einige der Html Seiten beinhalten soll.
[crayon-680c106ea6690845559482/]
Die Razor Anweisung @ViewBag.Title erlaubt es dann später jeder View einen anderen Titel vergeben zu können.

Die Anweisung @RenderBody() sagt aus, das hier der Inhalt der anderen Views stehen wird.

Im Controller kann man, wenn man nun eine neue View erstellt diese Template View auswählen:

Anschließend lässt sich im ViewBag.Title der Titel Der Seite anpassen und im Layout sehen wir wohin der Verweis führt.

Im unteren Block lässt sich nun einfach nur der gewünschte Inhalt anpassen
[crayon-680c106ea6698930525800/]

Sections

Mann kann aber auch Innerhalb von Template Seiten an

bestimmten Stellen Sektionen anfügen, die die Views mit eigenem Inhalt füllen soll.
[crayon-680c106ea669a720790777/]
und in der View:
[crayon-680c106ea669b079285316/]

ASP MVC View Razor Syntax

Eigentlich ist die Razor Syntax recht einfach gehalten, dennoch vergisst man sie schnell, wenn man nicht oft damit arbeitet.

- Einfachen Code Block erstellen: **@{ *C#* }**
- Variable ausgeben: **@VarName**
- String Zeichenkette ausgeben, ohne Variable(Ohne " "): **@:HalloWelt**
- Zeichenverkettung: **@var myVariable = „Baum“**
  @:ich habe einen schönen @(myVariable) der immer blüht.
- Textausgabe innerhalb des Code Blockes ohne HTML Tags: **@{ C# <Text> C# </Text>}**
- Textausgabe innerhalb des Code Blockes mit HTML Tags: **@{ *C#* <h1> *C#* </h1>}**
- Kommentar, der nicht in der HTML Seite sichtbar ist: **@* *mein Kommentar* *@**
- Normalerweise erkennt Razor automatisch, wenn es sich um eine E-Mail Adresse handelt und gibt diese auch so aus, falls dem nicht so ist, muss man das @ Zeichen mit **@@** Escapen, weil dieses Zeichen ja auch den Razor Code Block andeutet

Möchte man aus dem Controller etwas an die View übergeben, eignet sich der seit ASP 3 eingeführte dynamische Datentyp ViewBag. Dazu erstellt man im Controller einen neuen ViewBag: **ViewBag.VariablenName = „Wert“;**

und gibt diese in der View mit **@ViewBag.VariablenName** einfach aus. Ein casten ist hiermit nicht notwendig.