



# *Web Côté Serveur*

## *HTML Helpers*

## Présentation

*La classe HtmlHelper génère des balises html à l'aide de l'objet classe de modèle en Razor View. Il fait le binding de l'objet modèle aux éléments html pour afficher la valeur des propriétés du modèle en balises html et affecte également la valeur des éléments html aux propriétés du modèle lors de la l'envoi d'un formulaire Web. L'utilisation de la Classe HtmlHelper dans une View au lieu d'écrire manuellement les balises html.*

# Présentation

*La figure suivante montre l'utilisation de la classe HtmlHelper dans une View Razor :*

```
@model IEnumerable<MVC_BasicTutorials.Models.Student>

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.StudentName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Age)
        </th>
        <th></th>
    </tr>
```

HtmlHelper

Extension method for HtmlHelper

# Présentation

Comme vous pouvez le voir dans la figure ci-dessus, `@Html` est un objet de la classe `HtmlHelper`. (Le symbole `@` est utilisé pour accéder à l'objet côté serveur dans la syntaxe Razor). `Html` est une propriété de type `HtmlHelper` incluse dans la classe de base de Razor View `WebViewPage`.

`ActionLink()` et `DisplayNameFor()` sont des méthodes d'extension incluses dans la classe `HtmlHelper`.

La classe `HtmlHelper` génère des éléments html. Par exemple :

`@ Html.ActionLink ("Créer nouveau", " Create")` générerait une balise d'ancrage

`<a href="/Student/Create"> Créer nouveau </a>`.

# Méthodes HtmlHelper

*Le tableau suivant répertorie les méthodes HtmlHelper et le contrôle html généré par chaque méthode.*

Méthode d'extension	Méthode fortement typée	Contrôle HTML
Html.ActionLink ()	N / A	<a> </a>
Html.TextBox ()	Html.TextBoxFor ()	<input type = "textbox">
Html.TextArea ()	Html.TextAreaFor ()	<input type = "textarea">
Html.CheckBox ()	Html.CheckBoxFor ()	<input type = "checkbox">
Html.RadioButton ()	Html.RadioButtonFor ()	<input type = "radio">
Html.DropDownList ()	Html.DropDownListFor ()	<select> <option> </select>
Html.ListBox ()	Html.ListBoxFor ()	zone de liste à sélection multiple: <select>
Html.Hidden ()	Html.HiddenFor ()	<input type = "hidden">
Html.Password ()	Html.PasswordFor ()	<input type = "password">
Affichage Html ()	Html.DisplayFor ()	Texte HTML: ""
Html.Label ()	Html.LabelFor ()	<label>
Html.Editor ()	Html.EditorFor ()	Génère des contrôles Html basés sur le type de données de la propriété de modèle spécifiée, par exemple une zone de texte pour une propriété de chaîne, un champ numérique pour un int, un double ou un autre type numérique.

## *Méthodes HtmlHelper*

*La différence entre appeler les méthodes HtmlHelper et utiliser des balises html est que la méthode HtmlHelper est conçue pour faciliter la liaison pour afficher des données ou des données de modèle.*

## HtmlHelper – TextBox

*La classe HtmlHelper comprend deux méthodes d'extension qui créent un élément de zone de texte (<input type = "text">) dans une vue: TextBox () et TextBoxFor (). La méthode TextBox () est une méthode faiblement typée alors que TextBoxFor () est une méthode fortement typée.*

Example1: Html.TextBox() in Razor View

```
@model Student
```

```
@Html.TextBox("StudentName", null, new { @class = "form-control" })
```

Résultat Html:

```
<input class="form-control"
      id="StudentName"
      name="StudentName"
      type="text"
      value="" />
```

## HtmlHelper – TextBox

*Dans l'exemple ci-dessus, le premier paramètre est la propriété "StudentName" de la classe de modèle Student qui sera définie en tant que nom et id de la zone de texte. Le deuxième paramètre est une valeur à afficher dans une zone de texte, qui est nulle dans l'exemple ci-dessus car la méthode TextBox () affichera automatiquement une valeur de la propriété StudentName dans la zone de texte. Le troisième paramètre sera défini comme attribut de classe. Le paramètre HtmlAttributes est un type d'objet, il peut donc être un objet anonyme et le nom des attributs sera ses propriétés commençant par le symbole @.*

*Vous pouvez également spécifier n'importe quel nom pour la zone de texte. Cependant, il ne sera pas lié à un modèle.*



# HtmlHelper – TextBox

## Exemple2: Html.TextBox() en Razor View

```
@Html.TextBox("myTextBox", "This is value", new { @class = "form-control" })
```

### Result Html:

```
<input class="form-control"
      id="myTextBox"
      name="myTextBox"
      type="text"
      value="This is value" />
```

# TextBoxFor

*La méthode d'assistance TextBoxFor est une méthode d'extension fortement typée. Il génère un élément de saisie de texte pour la propriété de modèle spécifiée à l'aide d'une expression lambda. La méthode TextBoxFor lie une propriété d'objet de modèle spécifiée au texte d'entrée. Il affiche donc automatiquement une valeur de la propriété du modèle dans une zone de texte et vice-versa.*

## Exemple: TextBoxFor() en Razor View

```
@model Student
```

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

## Résultat HTML:

```
<input class="form-control"
      id="StudentName"
      name="StudentName"
      type="text"
      value="John" />
```

## *Différence entre TextBox et TextBoxFor*

*@ Html.TextBox () est une méthode est faiblement typé alors que*

*@ Html.TextBoxFor () est une méthode d'extension fortement typée (générique).*

*TextBox () requiert le nom de la propriété comme paramètre de chaîne alors que TextBoxFor () requiert l'expression lambda comme paramètre.*

*TextBox ne vous donne pas d'erreur de temps de compilation si vous avez spécifié un nom de propriété incorrect. Il lèvera une exception de temps d'exécution.*

*TextBoxFor est une méthode générique qui vous donnera une erreur en temps de compilation si vous avez spécifié un mauvais nom de propriété ou des changements de nom de propriété. (La vue fournie n'est pas compilée au moment de l'exécution.)*