**Html Helper Methods In Asp.Net Mvc 5**

**In ASP.NET MVC Framework, helper methods:**

* Are extension methods to the Html Helper class, can be called only from views.
* An HTML helper is a method that is used to render html content in a view.
* Simplifies the process of creating a view.
* Allows generating HTML markup that you can reuse across the Web application.

**Some of the commonly used helper methods while developing an MVC application are as follows:**

* Html.ActionLink()
* Html.BeginForm() and Html.EndForm()
* Html.Label()
* Html.TextBox()
* Html.TextArea()
* Html.Password()
* Html.CheckBox()

**Html.ActionLink()**

Html.ActionLink() helper method allows you to generate a hyperlink that points to an action method of a controller class.

The general syntax of the Html.ActionLink() helper method is as follows:

**@Html.ActionLink(<link\_text>,<action\_method>,<optional\_controller>)**

**HTML HELPER METHODS IN ASP.NET MVC 5**

An HTML helper is a method that is used to render html content in a view

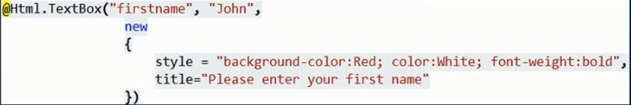
**Some of the commonly used helper methods while developing an MVC application are as follows:**

* Html.ActionLink()
* Html.BeginForm() and Html.EndForm()
* Html.Label()
* Html.TextBox()
* Html.TextArea()
* Html.Password()
* Html.CheckBox()
* To create the html for a textbox with id = “fullName” and name = “fullName”
  + **<input type=”text” name = “fullName” id = “fullName”>**

**OR**

* We can use the “TextBox” Html helper class method.
  + **@Html.TextBox(“fullName”)**
* There are several overloaded versions. To set a value, along with name of a textbox
  + **@Html.TextBox(“fullName”,”Adil”)**

To set HTML attributes, use the following overloaded version. Note that: We are passing HTML attributes (style and title) as an anonymous type.



* We can use external style sheet to give the style to the textbox.
* Applying bootstrap class to textbox.
* If you want to use attributes that are reserved in Csharp programming language like class, readonly etc. Then you have to use @ symbol before the name of attribute.

**HTML HELPER METHODS IN ASP.NET MVC 5**

**Some of the commonly used helper methods while developing an MVC application are as follows:**

* Html.ActionLink()
* Html.BeginForm() and Html.EndForm()
* Html.Label()
* Html.TextBox()
* Html.Password()
* Html.RadioButton()
* Html.DropDownList()
* Html.TextArea()
* Html.CheckBox()
* Html.hidden()
* Is it compulsory to use html helpers?
  + No, you can type the required html, but using html helpers will greatly reduce the amount of HTML markup that we have to write in a view. Views should be as simple as possible.

**In ASP.NET MVC Framework, helper methods:**

* HTML Helpers are extension methods to the Html Helper class, can be called only from views.
* An HTML helper is a method that is used to render html content in a view.
* HTML Helpers simplifies the process of creating a view.
* Allows generating HTML markup that you can reuse across the Web application.

**Some of the commonly used helper methods while developing an MVC application are as follows:**

* Html.ActionLink()
* Html.BeginForm() and Html.EndForm()
* Html.Label()
* Html.TextBox()
* Html.TextArea()
* Html.Password()
* Html.CheckBox()

**Html.ActionLink() Helper Method In ASP.Net MVC**

**Html.ActionLink()** helper method allows you to generate a **hyperlink** that points to an action method of a controller class.

**The general syntax of the Html.ActionLink() helper method is as follows:**

@Html.ActionLink(<link\_text>,<action\_method>,<optional\_controller>)

**where,**

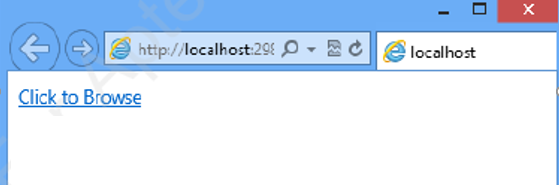
* **link\_text:** Is the text to be displayed as a hyperlink.
* **action\_method:** Is the name of the action method that acts as the target for the hyperlink
* **optional\_controller:** Is the name of the controller that contains the action method that will get called by the hyperlink. This parameter can be omitted if the action method getting called is in the same controller as the action method whose view renders the hyperlink.

**Following code snippet shows using a Html.ActionLink() helper method:**

@Html.ActionLink("Click to Browse", "Browse", "Home")

In the above code, the **Click to Browse** text will be displayed as a hyperlink.

The Browse action method of the HomeController acts as the target of the hyperlink, as shown in the following figure:



**Html.BeginForm() helper method:**

* Html.BeginForm() allows you to mark the start of a form.
* Html.BeginForm() co-ordinates with the routing engine to generate a URL.
* Html.BeginForm() is responsible for producing the opening
* tag.

**The general syntax of the Html.BeginForm() helper method is as follows:**

@{Html.BeginForm(<action\_method>,<controller\_name>);}

**where,**

* **action\_method:** Is the name of the action method.
* **controller\_name:** Is the name of the controller class.

Once you use the **Html.BeginForm()** helper method to start a form, you need to end a form using the **Html.EndForm()** helper method.

**Following code snippet shows using the Html.BeginForm() and Html.EndForm() helper methods:**

@{Html.BeginForm("Browse","Home");}

<p>Inside Form</p>

@{Html.EndForm();}

In the above code the **Html.BeginForm()** method specifies the Browse action of the HomeController as the target action method to which the form data will be submitted.

You can also avoid using the **Html.EndForm()** helper method to explicitly close the form by using the **@using** statement before the **Html.BeginForm()** method.

**Html.Label() helper method:**

* Allows you to display a label in a form.
* Enables attaching information to other input elements, such as text inputs, and increase the accessibility of your application.

**The general syntax of the Html.Label() helper method is as follows:**

@Html.Label(<label\_text>)

**where,**

**label\_text:** Is the name of the label.

**Html.TextBox() helper method:**

* Allows you to display an input tag.
* Used to accept input from a user.

**To create the html for a textbox with id = “fullName” and name = “fullName”**

<input type=”text” name = “fullName” id = “fullName”>

**OR**

**We can use the “TextBox” Html helper class method**

@Html.TextBox(“fullName”)

**There are several overloaded versions. To set a value, along with name of a textbox**

@Html.TextBox(“fullName”,”Adil”)

**To set HTML attributes, use the following overloaded version. Note that: We are passing HTML attributes (style and title) as an anonymous type.**



**Following code snippet shows using a Html.TextBox()method:**

@{Html.BeginForm("Browse","Home");}

@Html.Label("User Name:")</br>

@Html.TextBox("textBox1")</br></br>

<input type="submit" value="Submit">

@{Html.EndForm();}

**Html.TextArea() helper method:**

* Allows you to display a element for multi-line text entry.</p>
* Enables you to specify the number of columns and rows to be displayed in order to control the size of the text area.

**Following code snippet shows using a Html.TextArea() method:**

@{Html.BeginForm("Browse","Home");}

@Html.Label("User Name:")</br>

@Html.TextBox("textBox1")</br></br>

@Html.Label("Address:")</br>

@Html.TextArea("textarea1")</br></br>

<input type="submit" value="Submit">

@{Html.EndForm();}

**Html.Password() helper method:**

You can use the **Html.Password()** helper method to display a **password** field with bulleted text inside in place of original password.

**Following code snippet shows using a Html.Password() method:**

@{Html.BeginForm("Browse","Home");}

@Html.Label("User Name:")</br>

@Html.TextBox("textBox1")</br></br>

@Html.Label("Address:")</br> @Html.TextArea("textarea1")</br></br>

@Html.Label("Password:")</br>@Html.Password("password")</br></br>

<input type="submit" value="Submit">

@{Html.EndForm();}

**Html.CheckBox() helper method:**

You can use the **Html.CheckBox()** helper method to display a check box that enables the user to select a true or false condition.

**Following code snippet shows using a Html.CheckBox() method:**

@{Html.BeginForm("Browse","Home");}

@Html.Label("User Name:")</br>

@Html.TextBox("textBox1")</br></br>

@Html.Label("Address:")</br> @Html.TextArea("textarea1")</br></br>

@Html.Label("Password:")</br>@Html.Password("password")</br></br>

@Html.Label("I need updates on my mail:")

@Html.CheckBox ("checkbox1")</br> </br>

<input type="submit" value="Submit"> @{Html.EndForm();}

In this code, the Html.CheckBox() helper method renders a hidden input in addition to the check box input. The hidden input ensures that a value will be submitted, even if the user does not select the check box.

**Html.DropDownList() helper method:**

* Return a <**select>** element that shows a list of possible options and also the current value for a field.
* Allows selection of a single item.

**The general syntax of the Html.DropDownList() helper method is as follows:**

@Html.DropDownList("myList", new SelectList(new [] {<value1>, <value2>,

< value3>}), "Choose")

**where,**

* **value1**,**value2** and **value3** are the options available in the drop-down list.
* **Choose:** Is the value at the top of the list.

**Following code snippet shows using a Html.DropDownList() method:**

@{Html.BeginForm("Browse","Home");}

@Html.Label("User Name:")</br>

@Html.TextBox("textBox1")</br></br>

@Html.Label("Address:")</br> @Html.TextArea("textarea1")</br></br>

@Html.Label("Password:")</br>@Html.Password("password")</br></br>

@Html.Label("I need updates on my mail:")

@Html.CheckBox("checkbox1")</br> </br>

@Html.Label("Select your city:")

@Html.DropDownList("myList", new SelectList(new [] {"New York", "Philadelphia", "California"}), "Choose")</> </br></br>

<input type="submit" value="Submit">

@{Html.EndForm();}

In Above code, the **Html.DropDownList()** method creates a drop-down list in a form with myList as its name and contains three values that a user can select from the drop-down list.

**Html.RadioButton() helper method:**

The **Html.RadioButton()** helper method allows you to provide a range of possible options for a single value.

**The general syntax of the Html.RadioButton() helper method is as follows:**

@Html.RadioButton("name","value",isChecked)

**where,**

* **name:** Is the name of the radio button input element.
* **value:** Is the value associated with a particular radio button option.
* **isChecked:** Is a Boolean value that indicates whether the radio button option is selected or not.

@{Html.BeginForm("Browse","Home");}

@Html.Label("User Name:")</br>

@Html.TextBox("textBox1")</br></br>

@Html.Label("Address:")</br> @Html.TextArea("textarea1")</br></br>

@Html.Label("Password:")</br>@Html.Password("password")</br></br>

@Html.Label("I need updates on my mail:")@Html.CheckBox("checkbox1")</br> </br>

@Html.Label("Select your city:") @Html.DropDownList("myList", new SelectList(new [] {"New York", "Philadelphia", "California"}), "Choose")</> </br></br>

Male @Html.RadioButton("Gender", "Male", true)</br>

Female @Html.RadioButton("Gender", "Female")</br> </br>

<input type="submit" value="Submit">

@{Html.EndForm();}

In Above code, the Html.RadioButton() helper methods is used to create two radio buttons to accept the gender of a user.

**Url.Action() helper method:**

The **Url.Action()** helper method generates a URL that targets a specified action method of a controller.

**The general syntax of the Url.Action() helper method is as follows:**

@Url.Action(<action\_name>, <controller\_name>)

**where,**

* **action\_name:** Is the name of the action method.
* **controller\_name:** Is the name of the controller class.

**Following code snippet shows the Url.Action() method:**

<a href='@Url.Action("Browse", "Home")'>Browse</a>

Above code creates a hyperlink that targets the URL generated using the **Url.Action()** method. When a user clicks the hyperlink, the Browse action of the Home controller will be invoked.

**More Information About HTML Helpers**

* We can use external style sheet to give the style to the textbox.
* We can apply bootstrap class to textbox.
* If you want to use attributes that are reserved in Csharp programming language like class, readonly etc. Then you have to use @ symbol before the name of attribute.

**Is it compulsory to use html helpers in ASP.Net MVC Application?**

**No,** you can type the required html, but using html helpers will greatly reduce the amount of HTML markup that we have to write in a view. Views should be as simple as possible.