HtmlHelper - Editor

We have seen different HtmlHelper methods used to generated different html elements in the previous sections. ASP.NET MVC also includes a method that generates html input elements based on the datatype. Editor() or EditorFor() extension method generates html elements based on the data type of the model object's property.

The following table list the html element created for each data type by Editor() or EditorFor() method.

| Property DataType | Html Element |
| --- | --- |
| string | <input type="text" > |
| int | <input type="number" > |
| decimal, float | <input type="text" > |
| boolean | <input type="checkbox" > |
| Enum | <input type="text" > |
| DateTime | <input type="datetime" > |

We will use the following model class with Editor and EditorFor method.

Example: Student Model

public class Student

{

public int StudentId { get; set; }

[Display(Name="Name")]

public string StudentName { get; set; }

public int Age { get; set; }

public bool isNewlyEnrolled { get; set; }

public string Password { get; set; }

public DateTime DoB { get; set; }

}

Editor()

Editor() method requires a string expression parameter to specify the property name. It creats a html element based on the datatype of the specified property.

Editor() signature: MvcHtmlString Editor(string propertyname)

Visit MSDN to know all the overloads of Editor() method

Consider the following example to understand the Editor() method.

Example: Editor() in Razor view

StudentId: @Html.Editor("StudentId")

Student Name: @Html.Editor("StudentName")

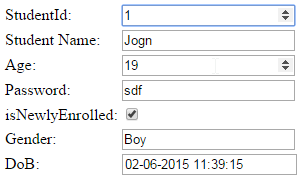
Age: @Html.Editor("Age")

Password: @Html.Editor("Password")

isNewlyEnrolled: @Html.Editor("isNewlyEnrolled")

Gender: @Html.Editor("Gender")

DoB: @Html.Editor("DoB")

Output of Editor() and EditorFor() method

In the above example, we have specified property names of Student model as a string. So, Editor() method created the appropriate input elements based on the datatype as shown in the above figure.

EditorFor

EditorFor() method is a strongly typed method. It requires the lambda expression to specify a property of the model object.

EditorFor() signature: MvcHtmlString EditorFor(<Expression<Func<TModel,TValue>> expression)

Visit MSDN to know all the overloads of EditorFor() method

Example: EditorFor() in Razor view

StudentId: @Html.EditorFor(m => m.StudentId)

Student Name: @Html.EditorFor(m => m.StudentName)

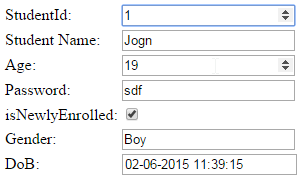
Age: @Html.EditorFor(m => m.Age)

Password: @Html.EditorFor(m => m.Password)

isNewlyEnrolled: @Html.EditorFor(m => m.isNewlyEnrolled)

Gender: @Html.EditorFor(m => m.Gender)

DoB: @Html.EditorFor(m => m.DoB)

Output of Editor() and EditorFor() method

In the above example of EditorFor() method, we have specified the property name using the lambda expression. The result would be the same as the Editor() method as shown in the above figure.