Html Helper

Lesson Objectives

- *HTML Helpers*
- *Build-in Helpers*
- *Create custom helper*
- *Helper extension*
- *Templated Helpers*

Section 1

HTML HELPERS

- With MVC, we can create view by using HTML code only
  - ✓ **<input type="text" name="ProductName" id="ProductName" />**
- It becomes complex, difficult to read, understand and update
  - ✓ Long code
  - ✓ Take long time to develop
  - ✓ Not strongly typed

HTML Helper

- <input type="text" name="ProductName" id="ProductName" />

is replaced by:

- @Html.TextBox("ProductName")

HTML Helper

- An HTML helper in MVC is an extension method of the HTML Helper class which is used to generate HTML content in a view.
- HtmlHelper method is designed to make it easy to bind to view data or model data.
- MVC use extension methods to create methods in HtmlHelper class

HTML Helper

- There are several overloaded versions available for each method in HTML helper
- It provides basic properties for common case
  - ✓ @Html.TextBox("ProductName")
  - ✓ @Html.TextBox("ProductName", "Apple Macbook")
  - ✓ @Html.TextBox("ProductName", "Apple Macbook", new { title ="Please enter product name" })

HTML Helper

- It is also possible to set the HTML attributes
  - ✓ passing the HTML attributes as an anonymous type
  - ✓ Some of the HTML attributes are reserved keywords, we use @ symbol for them
  - ✓ @Html.TextBox("ProductName", "Apple Macbook", **new** { title = "Please enter product name" , **@class** = "new-product", **@readonly** ="true" })

Section 2

BUILD-IN HELPERS

TextBox

- Used to generate HTML input, type is text (usually called textbox)
- There are 7 overloads in MVC

```
public static MvcHtmlString TextBox(this HtmlHelper htmlHelper, string name, object value, string format, IDictionary<string, object> htmlAttr
public static MvcHtmlString TextBox(this HtmlHelper htmlHelper, string name, object value, IDictionary<string, object> htmlAttributes);
public static MvcHtmlString TextBox(this HtmlHelper htmlHelper, string name, object value, string format, object htmlAttributes);
public static MvcHtmlString TextBox(this HtmlHelper htmlHelper, string name, object value);
public static MvcHtmlString TextBox(this HtmlHelper htmlHelper, string name, object value, string format);
public static MvcHtmlString TextBox(this HtmlHelper htmlHelper, string name);
public static MvcHtmlString TextBox(this HtmlHelper htmlHelper, string name, object value, object htmlAttributes);
```

TextBox

- Parameters:
  - ✓ name: The name of the form field and the System.Web.Mvc.ViewDataDictionary key that is used to look up the value
  - ✓ value: The value of the text input element.
  - ✓ format: A string that is used to format the input
  - ✓ htmlAttributes: An object that contains the HTML attributes to set for the element.

TextBoxFor

- Used to generate a text input element for the model property specified using a lambda expression.
- Binds a specified model object property to input text.
- It automatically displays a value of the model property in a textbox in both directions.

TextBoxFor

- Example:
  - ✓ @Html.TextBoxFor(model => model.ProductName, new { @class = "new-product" })
- Parameters:
  - ✓ expression: An expression that identifies the object that contains the properties to display.

TextBox vs. TextBoxFor

TextBox

- @Html.TextBox() is loosely typed method
- TextBox() requires property name as string parameter
- TextBox doesn't give you compile time error if you have specified wrong property name. It will throw run time exception.

TextBoxFor

- @Html.TextBoxFor() is a strongly typed (generic) extension method
- TextBoxFor() requires lambda expression as a parameter.
- TextBoxFor is generic method so it will give you compile time error if you have specified wrong property name or property name changes.

TextArea

- Used to generate HTML textarea with rows and cols attributes
- Example:
  - ✓ @Html.TextArea("Description", "This is newest Apple Macbook Pro 2019", 6, 10, new { @class = "new-product" })

TextArea

- Parameters:
  - ✓ name:
  - ✓ value
  - ✓ rows: The number of rows.
  - ✓ columns: The number of columns.
  - ✓ htmlAttribute

The rule:

- Each method should have a pair, 1 is loosely typed, 1 is strongly typed
- Strongly typed method is end with **For**

CheckBox- CheckBoxFor

- Used to generate input tag with type is checkbox
- Parameter isChecked to set the checkbox is checked by default or not

RadioButton- RadioButtonFor

- Used to generate input tag with type is radio.
- Radios are always come in group.
- All radios with same name are considered as one group
- Parameter isChecked to set the checkbox is checked by default or not

DropDownList- DropDownListFor

- Used to generate <select> tag.
- DropDownList requires a IEnumerable<SelectListItem>

- DropDownList vs RadioButton group
  - ✓ What are differences between them?
  - ✓ When do we use either DropDownList orRadioButton group?

Editor - EditorFor

- Generate HTML elements based on the data type of the model object's property

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" >

Others

- Hidden
- Password
- Display
- Label
- ValidationMessage

Section 3

CREATE CUSTOM HELPER

Section 4

HELPER EXTENSION

Section 5

TEMPLATED HELPERS

Lesson Summary

Thank you