**Data Annotations In Asp.Net Mvc**

The MVC Framework provides several data annotations that you can apply as attributes to a model.

These data annotations implement tasks that are commonly required across applications.

**Some of the important annotations that you can use in the models of an ASP.NET MVC application are as follows:**

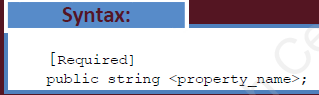
* Required
* StringLength
* RegularExpression
* Range
* Compare
* DisplayName
* ReadOnly
* DataType
* ScaffoldColumn

**Required Data Annotation**

The Required data annotation specifies that the property, with which this annotation is associated, is a required property.

This attribute raises a validation error if the property value is null or empty.

Following is the syntax of the Required data annotation:



**String Length Data Annotation In Mvc**

* You can use the StringLength annotation to specify the minimum and maximum lengths of a string field.
* Whenever a user specified values for these fields are out of this specified range a validation error message is displayed

**Following is the syntax for StringLength annotation:**

[StringLength(<max\_length>, MinimumLength= <min\_length>)]

**Regular Expression Data Annotation In Mvc**

You can use the RegularExpression annotation to accept user input in a specific format.

This annotation allows you to match a text string with a search pattern that contains one or more character literals, operators, or constructs.

Following is the syntax for RegularExpression annotation:

[RegularExpression(<pattern>)]

**Range Data Annotation In Mvc**

You can use the Range annotation to specify the minimum and maximum constraints for a numeric value.

Following is the syntax of the Range annotation:

[Range (<minimum\_range>, <maximum\_range>)]

where,

minimum\_range: Is a numeric value that specifies the minimum value for the range.

maximum\_range: Is a numeric value that specifies the maximum value for the range.

**COMPARE DATA ANNOTATION IN MVC**

* You can use the Compare annotation to compare values in two fields.
* The Compare annotation ensures that the two properties on a model object have the same value.

**DISPLAY NAME DATA ANNOTATION IN MVC**

When you use the @Html.LabelFor() helper method in a strongly typed view, the method displays a label with a text whose value is the corresponding property name.

You can also explicitly state the text that the @Html.LabelFor() method should display using the DisplayName annotation on the model property.

**READ ONLY DATA ANNOTATION IN MVC**

* You can use the ReadOnly annotation to display read-only fields on a form.
* You can use this annotation to instruct the default model binder not to set the specific property with a new value from the request.

Following is the syntax for ReadOnly annotation:

[ReadOnly(<boolean\_value>)]

where,

boolean\_value: Is a boolean value which can be either true or false. A true value indicates that the default model binder should not set a new value for the property. A false value indicates that the default model binder should set a new value for the property based on the request.

**DATATYPE DATA ANNOTATION IN MVC**

You can use the DataType annotation to provide information about the specific purpose of a property at run time.

Following is the syntax for DataType annotation:

[DataType(DataType.<value>)]

* Password
* MultilineText
* Date
* Time

**SCAFFOLD COLUMN DATA ANNOTATION**

When you use scaffolding using the Create template, the view by default will create UI fields for all the properties of the model.

However, you might need to ensure that the template does not create UI fields for certain properties.

In such scenarios, you can use the ScaffoldColumn annotation passing a false value.