Data Annotation Attributes in ASP.NET MVC

By Simplilearn

Last updated on Jul 23, 2024   👁 38320

Share This Article: ⧉
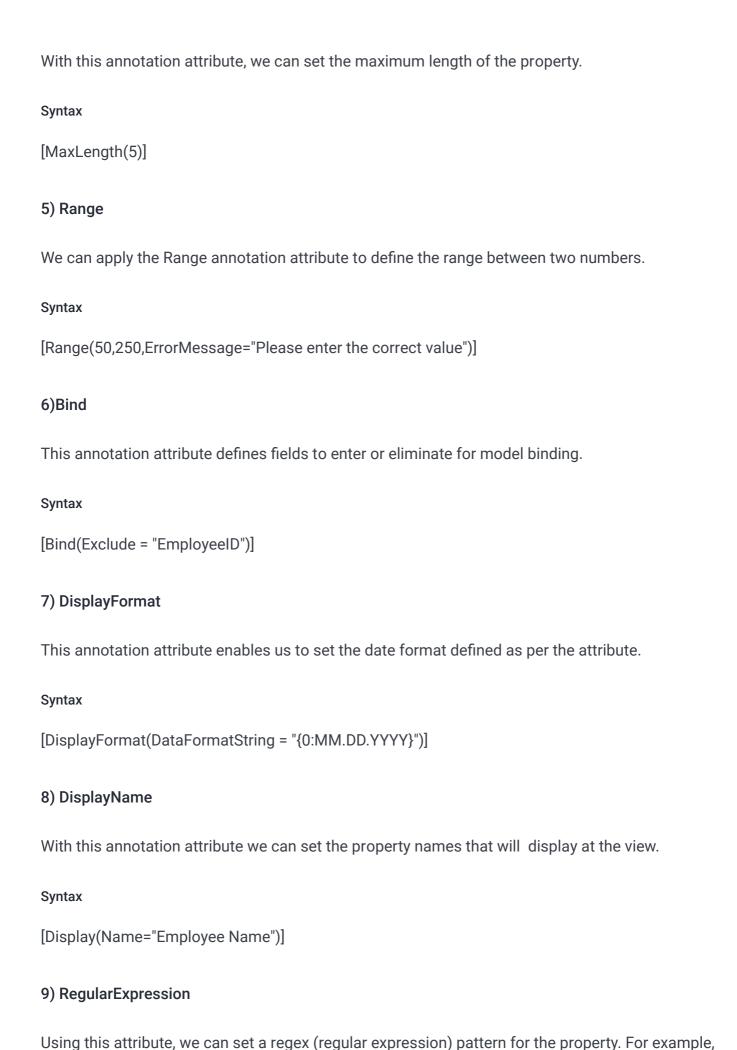

Table of Contents   ▾

In ASP.NET MVC, Data Annotation is used for data validation for developing web-based applications. We can quickly apply validation with the help of data annotation attribute classes over model classes. Data Annotations accommodate us to establish the controls to the model properties or classes for data validation and performing applying messages to users. In this article, we will understand the concepts of data annotation in MVC.

Data Annotation Attributes in ASP.NET MVC

Here are the numerous types of Data Annotations with the syntax:

1) DataType

This attribute is used to specify the data type of the model.

Syntax

[DataType(DataType.Text)]

2) Required

This attribute defines the particular value as mandatory for a certain requirement

Syntax

[Required(ErrorMessage="Please enter your name"),MaxLength(50)]

3) StringLength

Using this annotation attribute, we can set the maximum and minimum string range of the property.

Syntax

[StringLength(50,ErrorMessage="Please do not enter values over 50 characters")]

4) MaxLength

With this annotation attribute, we can set the maximum length of the property.

Syntax

[MaxLength(5)]

5) Range

We can apply the Range annotation attribute to define the range between two numbers.

Syntax

[Range(50,250,ErrorMessage="Please enter the correct value")]

6)Bind

This annotation attribute defines fields to enter or eliminate for model binding.

Syntax

[Bind(Exclude = "EmployeeID")]

7) DisplayFormat

This annotation attribute enables us to set the date format defined as per the attribute.

Syntax

[DisplayFormat(DataFormatString = "{0:MM.DD.YYYY}")]

8) DisplayName

With this annotation attribute we can set the property names that will display at the view.

Syntax

[Display(Name="Employee Name")]

9) RegularExpression

Using this attribute, we can set a regex (regular expression) pattern for the property. For example,

Email ID.

Syntax

[RegularExpression(@"^\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*$", ErrorMessage = "Your Email is not valid.")]

10) ScaffoldColumn

Specifies whether scaffolding is used by a class or a data column.

Syntax

[System.AttributeUsage(System.AttributeTargets.Field | System.AttributeTargets.Property, AllowMultiple=false)]

public class ScaffoldColumnAttribute : Attribute

In Data Annotations, we have two types of namespaces that have their specific in-build types.

1. System.ComponentModel.DataAnnotations incorporate the subsequent attributes that affect and check the size or nullability of the column.

- MinLength

- MaxLength

- StringLength

- Key

- Timestamp

- ConcurrencyCheck

- Required

2. System.ComponentModel.DataAnnotations.Schema namespace comprises the subsequent attributes that reshape the schema of the database.

- Index

- ForeignKey

- NotMapped

- InverseProperty

- Table

- Column

Why Do We Need Data Annotation Attributes in ASP.NET MVC?

In ASP.NET MVC web applications, we have the three types of data validations:

Client-Side Validation: HTML/JavaScript validation

Server-side Validation:

- ASP.NET MVC Model validation

- Database validation

Here we have ASP.NET MVC model validation that is a more secure validation whereas HTML/JavaScript validation is unsafe and can break after disabling the javascript while running the application.

ASP.NET MVC Framework implements a framework described as Data Annotation which is utilized for model validation. It is derived from System.ComponentModel.DataAnnotations assembly.

We can use Data Annotation to implement the model validation and can utilize it to model class properties.

What Are the Validations?

Validation is the set of rules which we define on the input fields on the webform page. We have certain types of validations which we can use as per the user requirements:

1. Client-Side Validation

2. Server-side Validation

In ASP.NET MVC, we have a System.ComponentModel.DataAnnotations assembly which has several built-in validation types of attributes' properties which we can implement as per the code logic.

- Required

- Data type

- RegularExpression

- Compare

- StringLength

- Range

- Credit Card number

- Currency

- Multi-line text

- Custom

- Date or DateTime

- Duration

- Email Address

- HTML

- Image URL

- Password

- Upload

- Phone number

- Postal Code

Built-in Validation Attributes

In ASP.NET MVC, we can consider Built-in validation attributes as custom validation. Below are a few examples:

- Compare: It validates and matches two fields' properties in a model.

- Range: It validates when a certain property value comes into a certain range.

- RegularExpression: It validates when a property value meets a certain regular expression.

- Remote: It works on a remote validation; we can call an action method function on the server and validate specific inputs on the client's side.

- DevExtremeRequired: It validates when the certain boolean property value is true or false.

- Custom : It allows you to define a custom validation attribute.

Want a Top Software Development Job? Start Here!

Full Stack Developer - MERN Stack

EXPLORE PROGRAM

Enabling Client-Side Validation in ASP.NET MVC Application

To enable the client-side validation, we can use JQuery and Javascript. Before using it, we should take the reference of the JavaScript files from the Scripts folder of your .net solution.

Example: jquery.validate.unobtrusive.js and query.validate.min.js

Below is the layout example:

<!DOCTYPE html>

<html>

<head>

   <meta charset="utf-8" />

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <title>@ViewBag.Title - Example Data annotation in MVC Application</title>

   @Styles.Render("~/Content/css")

   @Scripts.Render("~/bundles/modernizr")

</head>

<body>

   @Html.Partial("_HeaderNavBar");

   <div class="container body-content">

     @RenderBody()

     <hr />

```
    <footer>

        <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>

    </footer>

</div>

@Scripts.Render("~/bundles/jquery")

@Scripts.Render("~/Scripts/jquery.validate.min.js")

@Scripts.Render("~/Scripts/jquery.validate.unobtrusive.min.js")

@Scripts.Render("~/bundles/bootstrap")

@RenderSection("scripts", required: false)

</body>

</html>
```

Advance your career as a MEAN stack developer with the Full Stack Web Developer - MEAN Stack Master's Program. Enroll now!

Conclusion

We hope this article helped you understand Data Annotation in MVC using C#. In this article, we discussed the concept of data annotation and its various types, and validations with examples that will be helpful to professional developers from Java and .net backgrounds, application architectures, and other learners looking for information on JavaScript events.

Want a Top Software Development Job? Start Here!

Full Stack Developer - MERN Stack

Besides pursuing varied courses provided by Simplilearn, you can also sign up on our SkillUp platform. This platform, a Simplilearn initiative, offers numerous free online courses to help with the basics of multiple programming languages, including JavaScript. You can also opt for our Full-Stack Web Development Certification Course to improve your career prospects.

Our Software Development Courses Duration And Fees

Software Development Course typically range from a few weeks to several months, with fees varying based on program and institution.

Professional Certificate Program in Blockchain

Cohort Starts: **1 Jan, 2025**

4 Months	**$ 2,499**

Automation Test Engineer Masters Program

Cohort Starts: **18 Dec, 2024**

8 months	**$ 1,099**

Full Stack Java Developer Masters Program

Cohort Starts: **18 Dec, 2024**

7 months	**$ 1,500**

Full Stack (MERN Stack) Developer Masters Program

Cohort Starts: **8 Jan, 2025**

6 Months	**$ 1,500**

Recommended Reads

Data Science Career Guide: A Comprehensive Playbook To... Becoming A Data Scientist

12 Jun, 2023

What Is Image Annotation and Why Is It Important in Machine Learning

👁 1359

13 Aug, 2024

SOFTWARE DEVELOPMENT

ASP.NET MVC

👁 30867

29 Jul, 2024