



Data Annotations

.NET

Data Annotations Attributes enable you to perform validation simply by adding one or more attributes to a Model class property.

[HTTPS://DOCS.MICROSOFT.COM/EN-US/ASPNET/MVC/OVERVIEW/OLDER-VERSIONS-1/MODELS-DATA/VALIDATION-WITH-THE-DATA-ANNOTATION-VALIDATORS-CS](https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/models-data/validation-with-the-data-annotation-validators-cs)

Data Annotations – Overview

<https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/models-data/validation-with-the-data-annotation-validators-cs>

The *DataAnnotations* namespace provides a set of built-in **validation attributes** that are applied declaratively to a **class** or **property**.

DataAnnotations also contains formatting **attributes** like *DataType* that help with formatting but don't provide **validation**.

```
[StringLength(60, MinimumLength = 3)]
[Required]
public string Title { get; set; }

[Display(Name = "Release Date")]
[DataType(DataType.Date)]
public DateTime ReleaseDate { get; set; }

[Range(1, 100)]
[DataType(DataType.Currency)]
[Column(TypeName = "decimal(18, 2)")]
public decimal Price { get; set; }
```

Annotations to Display and Edit

[Display()]	[DataType()]
<code>[Display(Name = "Film Genre")]</code> This gives the View the desired name for Display in the browser	<code>[DataType(DataType.Password)]</code> Displays the types' text as dots to obscure the actual content.

```
[Display(Name = "Release Date")]
[DataType(DataType.Date)]
[DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
public DateTime ReleaseDate { get; set; }
```

Annotations for Validation

<https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/adding-validation>

Annotation	Explanation
[Required]	The property is not allowed to be null
[Range(x,y)]	X is the minimum. Y is the maximum.
[StringLength(x)]	X is the maximum length of the property
[RegularExpression]	[RegularExpression(@"^[A-Z]+[a-zA-Z0-9""'\s-]*\$")]
[MinLength(x)]	Minimum length is x. Also sets DB column size min.
[MaxLength(y, ErrorMessage- "This is required")]	Maximum length is y and this error message is displayed. Also sets DB column size max.

*Custom validation is also possible.

Data Annotations – Examples

<https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/adding-validation>

```
public class Movie
{
    public int ID { get; set; }

    [StringLength(60, MinimumLength = 3)]
    public string Title { get; set; }

    [Display(Name = "Release Date")]
    [DataType(DataType.Date)]
    [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
    public DateTime ReleaseDate { get; set; }

    [RegularExpression(@"^[A-Z]+[a-zA-Z'\s]*$")]
    [Required]
    [StringLength(30)]
    public string Genre { get; set; }

    [Range(1, 100)]
    [DataType(DataType.Currency)]
    public decimal Price { get; set; }

    [RegularExpression(@"^[A-Z]+[a-zA-Z'\s]*$")]
    [StringLength(5)]
    public string Rating { get; set; }
}
```

EF Code First Data Annotations

<https://docs.microsoft.com/en-us/ef/ef6/modeling/code-first/data-annotations#the-model>

Data Annotation	Explanation
[Key]	Annotates a property as the Key of the entity .
[Column(order=2)]	Used with [Key] to create a composite column. This will be the 2 nd key.
[ForeignKey("FK_ModelName")]	Marks a certain model as the FK for this model.
[Required]	This property will be required in the Db and client-side.
[NotMapped]	This property will not be mapped to the Db.
[ComplexType]	This annotation is placed on a subtype of a model to alert EF that the property on the model has properties of it own.
[ConcurrencyCheck]	Checks of changes between .SaveChanges() calls.
[Table("TableName")]	Placed above the Model Class name. Allows you to change the name of the table in the Db.
[Column("ColumnName")]	Allows you to name a column other than the property name.

*There are many more available

Custom Annotations

todo

Web API Annotations

todo