

How to create Custom Data Annotation Validators

Wanting to create custom data annotation validation. Are there any useful guides / samples on how to create them?

39

Firstly:

StringLength with minimum and maximum length. I'm aware .NET 4 can do this, but want to do the same in .NET 3.5, if possible being able to define minimum length only (at least x chars), maximum length only (up to x chars), or both (between x and y chars).

★

13

Secondly:

Validation using modulus arithmetic - if the number is a valid length, I wish to validate using the Modulus 11 algorithm (I have already implemented it in JavaScript, so I guess it would just be a simple porting?)

Update:

Solved second problem, was just a case of copying over the JavaScript implementation and making a few tweaks, so don't need a solution for that.

[asp.net](#)[asp.net-mvc-2](#)[data-annotations](#)

edited Aug 16 '12 at 18:26



Rap

5,146

1

38

76

asked Aug 5 '10 at 10:13



SamWM

2,429

9

51

80

2 Answers

▲

To create a custom data annotation validator follow these guidelines:

87

1. Your class has to inherit from `System.ComponentModel.DataAnnotations.ValidationAttribute` class.
2. Override `bool IsValid(object value)` method and implement validation logic inside it.

▼

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

 Google

Facebook 

Sometimes developers check that value is not null/empty and return false. This is *usually* incorrect behaviour, because that's on `Required` validator to check which means that your custom validators should only validate non-null data but return `true` otherwise (see example). This will make them usable on mandatory (required) and non-mandatory fields.

Example

```
public class StringLengthRangeAttribute : ValidationAttribute
{
    public int Minimum { get; set; }
    public int Maximum { get; set; }

    public StringLengthRangeAttribute()
    {
        this.Minimum = 0;
        this.Maximum = int.MaxValue;
    }

    public override bool IsValid(object value)
    {
        string strValue = value as string;
        if (!string.IsNullOrEmpty(strValue))
        {
            int len = strValue.Length;
            return len >= this.Minimum && len <= this.Maximum;
        }
        return true;
    }
}
```

All properties can be set in attribute as you wish to set them.
Some examples:

```
[Required]
[StringLengthRange(Minimum = 10, ErrorMessage = "Must be >10 characters.")]

[StringLengthRange(Maximum = 20)]

[Required]
```

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

 Google

Facebook 

Important

So this validator will still work on your model value that's not required, but when it's present it validates (think of a text field in a web form, that's not required, but if a user enters a value in, it has to be valid).

edited Feb 24 '14 at 16:59



demoncodemonkey

8,893 6 48 89

answered Aug 5 '10 at 10:39



Robert Koritnik

78.6k 43 246 368

so, in case of null value should isValid return true? – onof Aug 5 '10 at 10:53

So how do you add extra options, e.g. [MyValidator(Min=1, Max=20)] , Where both Min/Max (or just one) are optional? – SamWM Aug 5 '10 at 11:07

1 @onof: Yes. In that case it should return true . – Robert Koritnik Aug 5 '10 at 11:58

@Sam: I edited my answer and included an example of exactly what you're trying to do in the first scenario. – Robert Koritnik Aug 5 '10 at 12:08

@Rap it works for me in MVC 5. [msdn.microsoft.com/en-us/library/cc679289\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/cc679289(v=vs.110).aspx) – demoncodemonkey Feb 25 '14 at 8:26

Use the [CustomValidationAttribute](#) together with a validate function with signature

4 `public static ValidationResult Validate(MyType x, ValidationContext context)`

Example (for a string property)

```
using System.ComponentModel.DataAnnotations;
```

```
public class MyClass
```

```
{
```

```
    [CustomValidation(typeof(MyClass), "Validate")]
```

```
    public string MyProperty { get; set; }
```

```
    public static ValidationResult Validate(string x, ValidationContext context)
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

Google

Facebook

answered Jul 13 '17 at 9:07

[Micha Wiedenmann](#)**10.9k** 15 65 107

What is the ValidationContext context argument used for? – [petryuno1](#) Feb 22 '18 at 21:45

@petryuno1 You could use it to get the model if you happen to need to look at other properties in your validation. e.g., var model = (MyClass)context.ObjectInstance; – [Jason Butera](#) Jun 8 '18 at 15:22

Join **Stack Overflow** to learn, share knowledge, and build your career.

[Email Sign Up](#)

OR SIGN IN WITH

[Facebook](#)