MVC custom validation: compare two dates

Asked 8 years, 6 months ago Active 5 years ago Viewed 29k times

62

31

I've created a custom ValidationAttribute that compares 2 dates and makes sure that the second date is greater than the first:

```
public sealed class IsDateAfter : ValidationAttribute, IClientValidatable
{
    private readonly string testedPropertyName;
    private readonly bool allowEqualDates;

    public IsDateAfter(string testedPropertyName, bool allowEqualDates = false)
    {
        this.testedPropertyName = testedPropertyName;
        this.allowEqualDates = allowEqualDates;
    }

    protected override ValidationResult IsValid(object value, ValidationContext
validationContext)
    {
        var propertyTestedInfo =
validationContext.ObjectType.GetProperty(this.testedPropertyName);
        if (propertyTestedInfo == null)
        {
            return new ValidationResult(string.Format("unknown property {0}",
this.testedPropertyName));
        }

        var propertyTestedValue =
propertyTestedInfo.GetValue(validationContext.ObjectInstance, null);

        if (value == null || !(value is DateTime))
        {
            return ValidationResult.Success;
        }

        if (propertyTestedValue == null || !(propertyTestedValue is DateTime))
        {
```

```
            // Compare values
            if ((DateTime)value >= (DateTime)propertyTestedValue)
            {
                if (this.allowEqualDates)
                {
                    return ValidationResult.Success;
                }
                if ((DateTime)value > (DateTime)propertyTestedValue)
                {
                    return ValidationResult.Success;
                }
            }

            return new ValidationResult(FormatErrorMessage(validationContext.DisplayName));
        }

        public IEnumerable<ModelClientValidationRule> GetClientValidationRules(ModelMetadata
    metadata, ControllerContext context)
        {
            var rule = new ModelClientValidationRule
            {
                ErrorMessage = this.ErrorMessageString,
                ValidationType = "isdateafter"
            };
            rule.ValidationParameters["propertytested"] = this.testedPropertyName;
            rule.ValidationParameters["allowequaldates"] = this.allowEqualDates;
            yield return rule;
        }
```

CalendarEntry class: ...

```
public virtual DateTime StartDate { get; set; }

[IsDateAfter("StartDate", true, ErrorMessage="End date needs to be after start date")]
public virtual DateTime EndDate { get; set; }
```

VIEW:

```
$.validator.unobtrusive.adapters.add(
    'isdateafter', ['propertytested', 'allowequaldates'], function (options) {
    options.rules['isdateafter'] = options.params;
    options.messages['isdateafter'] = options.message;
});
```

```
    if (!value || !startdatevalue) return true;
    return (params.allowequaldates) ? Date.parse(startdatevalue) <= Date.parse(value) :
Date.parse(startdatevalue) < Date.parse(value);
}, '');
```

This works fine when the **CalendarEntry** is not wrapped inside another class. HOWEVER, when I use a view model like so:

```
    public class TrainingDateEditViewModel
    {
        #region Properties

        /// <summary>
        /// Gets or sets CalendarEntry.
        /// </summary>
        public CalendarEntry CalendarEntry { get; set; }
....
```

The client validation no longer works because the html output produced is this:

```
<input type="text" value="" name="CalendarEntry.EndDate" id="CalendarEntry_EndDate"
data-val-isdateafter-propertytested="StartDate" data-val-isdateafter-
allowequaldates="True" data-val-isdateafter="End date needs to be after start date"
data-val="true">
```

And the

```
data-val-isdateafter-propertytested="StartDate" and IT SHOULD BE:
"CalendarEntry.StartDate".
```

How would I make it so that it would know to bind to "CalendarEntry.StartDate" rule.ValidationParameters["propertytested"] = this.testedPropertyName; // HERE IT SHOULD BE FULL NAME??? HOW??

thanks

jquery asp.net-mvc asp.net-mvc-3 validation

edited Oct 22 '14 at 13:07 asked Aug 11 '11 at 11:40

2 thank you for this code, I've successfully implemented it in my validation – Sam Oct 7 '11 at 11:40

Thank you as well, for this, and @counsellorben for the update. – Andrew Backer May 16 '12 at 6:28

2 Might be a good idea to Suffix your class with `Attribute` – Liam Feb 5 '13 at 14:07

4 Answers

31

You need to modify your client-side script to check for the tested element's prefix, and add the prefix (if any) to your selector, as follows:

```
$.validator.addMethod("isdateafter", function(value, element, params) {
    var parts = element.name.split(".");
    var prefix = "";
    if (parts.length > 1)
        prefix = parts[0] + ".";
    var startdatevalue = $('input[name="' + prefix + params.propertytested +
'"]').val();
    if (!value || !startdatevalue)
        return true;
    return (params.allowequaldates) ? Date.parse(startdatevalue) <= Date.parse(value) :
        Date.parse(startdatevalue) < Date.parse(value);
});
```

answered Aug 11 '11 at 13:09

counsellorben
10.5k 3 34 36

4

Do not forget to include the client side inside this code. It tooks me hours to find that this was missing!

```
(function ($) {

  // your code here..

})(jQuery);
```


2,856 3 17 37


225 3 8

1

Just to fix a small error in counsellorben's javascript: the "(params.allowequaldates)" will be interpreted as a string (which will have a value of "False" or "True"), but that string will always be evaluated to true, thus always allowing equal dates. If you also want to allow for more levels of nesting of your objects than just 1, then you will get:

```
$.validator.addMethod("isdateafter", function(value, element, params) {
    var parts = element.name.split(".");
    var prefix = "";
    for (var i = 0; i < parts.length - 1; i++)
       prefix = parts[i] + ".";
    var startdatevalue = $('input[name="' + prefix + params.propertytested +
'"]').val();
    if (!value || !startdatevalue)
        return true;
    var allowequal = params.allowequaldates.toLowerCase === "true";
    return allowequal ? Date.parse(startdatevalue) <= Date.parse(value) :
        Date.parse(startdatevalue) < Date.parse(value);
});
```

edited Apr 28 '13 at 20:25

answered Apr 28 '13 at 19:50


gerard789
11 2

0

In the last answer there were some missing parenthesis on the call to toLowerCase, here is an updated version with document ready and the $.validator.unobtrusive...-part:

```
$(function () {
    $.validator.addMethod("isdateafter", function(value, element, params) {
        var parts = element.name.split(".");
        var prefix = "";
        for (var i = 0; i < parts.length - 1; i++) {
            prefix = parts[i] + ".";
        }

        var startdatevalue = $('input[name="' + prefix + params.propertytested +
```

```
            var allowequal = params.allowequaldates.toLowerCase() === "true";
            return allowequal ? Date.parse(startdatevalue) <= Date.parse(value) :
                Date.parse(startdatevalue) < Date.parse(value);
        });
        $.validator.unobtrusive.adapters.add('isdateafter',
            ['propertytested', 'allowequaldates'],
            function (options) {
                options.rules['isdateafter'] = options.params;
                options.messages['isdateafter'] = options.message;
            });
});
```

edited Aug 25 '14 at 17:52

answered Aug 25 '14 at 17:46

Dan Pettersson

663 4 15