# DRY principle and ViewModel data annotations [Answered]

2 replies

Last post Mar 08, 2016 08:09 AM by Mikesdotnetting

## DRY principle and ViewModel data annotations

Mar 07, 2016 11:07 PM | AceHigh26

I am new to MVC and I have a model library which are basically my DTO's and I have data annotations on the properties in each of my models. When I bind a model to my view I get nice clean validation out of the box. However I have been reading about "ViewModels" and I understand the purpose of them quite well. However one thing I am struggling with is the fact that for each of my ViewModels i am essentially repeating all of my data annotations again on the viewmodel properties. To me this goes against any sort of DRY principle and it doesnt seem right. What am I missing? It seems kind of messy to have the same annotations (regex and required field etc) repeated. Thanks

## Re: DRY principle and ViewModel data annotations

Mar 08, 2016 07:32 AM | Chris Zhao

Hi AceHigh26,

In ASP.NET MVC, ViewModels allow you to shape multiple entities from one or more data models or sources into a single object, optimized for consumption and rendering by the view. Take a look at **Use ViewModels to manage data & organize code in ASP.NET MVC applications** (http://rachelappel.com/use-viewmodels-to-manage-data-amp-organize-code-in-asp-net-mvc-applications/) . Not sure how you design your ViewModels, I usually use ViewModels like the following.

**Model**

```
public class DemoViewModel
    {
        public ModelA MA { get; set; }
        public ModelB MB { get; set; }
    }
```

```
public class ModelA
    {
        [Required]
        public int MAProperty1 { get; set; }
        [StringLength(5)]
        public string MAProperty2 { get; set; }
    }

public class ModelB
    {
        [Required]
        public int MBProperty1 { get; set; }
        [Required]
        public string MBProperty2 { get; set; }
    }
```

**Controller**

```
public ActionResult Index()
        {
            var model = new DemoViewModel();
            return View(model);
        }

        [HttpPost]
        public ActionResult Index(DemoViewModel model)
        {
            return View(model);
        }
```

**View**

```
@model CustomValidationDemo.Models.DemoViewModel
@{
    Layout = null;
}
```

```html
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
    <script type="text/javascript" src="~/Scripts/jquery-1.10.2.min.js"></script>
    <script type="text/javascript" src="~/Scripts/jquery.validate.min.js"></script>
    <script type="text/javascript" src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>
</head>
<body>
    <div>
        @using (@Html.BeginForm("Index", "Demo"))
        {
            <fieldset>
                @Html.TextBoxFor(m => m.MA.MAProperty1)
                @Html.ValidationMessageFor(m => m.MA.MAProperty1)
            </fieldset>
            <fieldset>
                @Html.TextBoxFor(m => m.MA.MAProperty2)
                @Html.ValidationMessageFor(m => m.MA.MAProperty2)
            </fieldset>
            <fieldset>
                @Html.TextBoxFor(m => m.MB.MBProperty1)
                @Html.ValidationMessageFor(m => m.MB.MBProperty1)
            </fieldset>
            <fieldset>
                @Html.TextBoxFor(m => m.MB.MBProperty1)
                @Html.ValidationMessageFor(m => m.MB.MBProperty1)
            </fieldset>
            <button type="submit">Submit</button>
        }
    </div>
</body>
</html>
```

Best Regards,

Chris Zhao

## Re: DRY principle and ViewModel data annotations

Mar 08, 2016 08:09 AM | Mikesdotnetting

I don't use annotations for validating user input at all on my domain entities/model. I only add those kinds of annotations to viewmodels.