عنوان: **ایجاد اعتبار سنجی های شرطی با Foolproof**
نویسنده: عثمان رحیمی
تاریخ: ۱۳۹۴/۰۳/۱۶ ۱۱:۳۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Validation, Data Annotation

ابتدا کلاس زیر را در نظر بگیرید:

```
public class UserVM
 {
        public string Name { get; set; }
        public bool  Gender { get; set; }
        public string Soldier { get; set; }
    }
```

قصد داریم یک سری اعتبار سنجی را بر روی خصوصیات کلاس فوق ایجاد کنیم. می‌خواهیم اگر کاربر جنسیت مرد را انتخاب کرد، حتما مقداری برای فیلد محل خدمت خود که در این کلاس Soldier می‌باشد، انتخاب کند. شاید انتخاب اول برای انجام چنین کاری، کنترل کردن آن در سمت کاربر با استفاده از جاوا اسکریپت باشد که می‌بایست یک رویداد را برای چک باکس جنسیت تعریف کنیم و بر اساس اینکه مرد انتخاب شده یا زن، ادامه کار را انجام دهیم.

روش اول: نوشتن یک کلاس سفارشی برای اعتبار سنجی کلاس فوق

```
public class SoldierValidation : ValidationAttribute
{
    public override bool IsValid(object value)
    {
        UserVM app = value as UserVM ;
        if (app.Gender && app.Soldier.Length==0)
        {
            ErrorMessage = "لطفا محل خدمت را وارد نمایید";

            return false;
        }
        return true;
    }
}
```

و سپس اعمال به کلاس مورد نظر همانند زیر :

```
[SoldierValidation ]
   public class UserVM
    {
        public string Name { get; set; }
        public bool  Gender { get; set; }
        public string Soldier { get; set; }
    }
```

تا اینجای کار، اگر کاربر از DropDown و یا RadioButton، آقا را انتخاب کرده باشد و View مورد نظر را برای Update و یا Insert ارسال کند، با خطای «لطفا محل خدمت را وارد نمایید» مواجه خواهد شد. تا به اینجا به مقصود مورد نظرمان رسیدیم.

روش دوم

لازم نیست چرخ رو دوباره اختراع کنید (البته در بعضی مواقع لازم است)

استفاده از Foolproof MVC:

Foolproof یک سری Annotation هایی را در اختیار شما قرار می‌دهد که با استفاده از آنها می‌توانید اعتبار سنجی‌های شرطی را انجام دهید؛ دقیقا همانند کاری که در بالا برای آن یک Validation سفارشی نوشتیم. البته Foolproof فقط به این مورد ختم نمی‌شود. در ادامه با چند مورد از آنها آشنا خواهیم شد.

ابتدا از طریق NuGet اقدام به نصب **Foolproof** نمایید:

```
PM> Install-Package foolproof
```

سپس اینبار همان مثال خود را با FoolPfoof انجام می‌دهیم:

```
public class UserVM
    {
        public string Name { get; set; }
        public bool  Gender { get; set; }
        [RequiredIfTrue("Gender ")]
        public string Soldier { get; set; }
    }
```

با استفاده از RequiredIfTrue دقیقا به همان مقصود خواهیم رسید که از ورودی، اسم فیلدی را می‌گیرد که می‌خواهیم آن را چک کنیم.

حال بپردازیم به چندین Annotation دیگر که در Foolproof وجود دارند:

GreatThan: همانطور که از نام آن پیداست، برای موقعیکه می‌خواهیم این فیلد بزرگتر از فیلد مورد نظرمان باشد:

```
public class EventViewModel
{
    public string Name { get; set; }
    public DateTime Start { get; set; }

    [Required]
    [GreaterThan("Start")]
    public DateTime End { get; set; }
}
```

جهت آشنایی بیشتر، در ادامه فقط لیست Annotationهای موجود در این پکیج قرار داده شده است .

```
[Is]
[EqualTo]
[NotEqualTo]
[GreaterThan]
[LessThan]
[GreaterThanOrEqualTo]
[LessThanOrEqualTo]
```

و

```
[RequiredIf]
[RequiredIfNot]
[RequiredIfTrue]
[RequiredIfFalse]
[RequiredIfEmpty]
[RequiredIfNotEmpty]
[RequiredIfRegExMatch]
[RequiredIfNotRegExMatch]
```

نظرات خوانندگان

نویسنده: مرتضی
تاریخ: ۱۳۹۴/۰۳/۱۶ ۱۲:۳۸

تشکر از مطلبتون

foolproof خیلی وقته بروزرسانی نشده

بهتره از https://github.com/JeremySkinner/FluentValidation استفاده کرد