عنوان: **تبدیل خودکار استثنای HttpRequestValidationException به یک ModelError در ASP.NET MVC**
نویسنده: سالمیان
تاریخ: ۱۳۹۴/۰۴/۳۰ ۱:۴۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security, Validation

فرم هایی که اطلاعاتی را از یک کاربر دریافت کرده و به سمت سرور Post می‌کنند، از مهمترین اجزای لاینفک یک وب سایت می‌باشند. بی شک همه‌ی ما از چنین فرمهایی حتی در یک پروژه‌ی هرچند کوچک استفاده کرده‌ایم. ممکن است هنگام ارسال این فرمها، کاربری شیطنت به خرج داده و درون یکی از فیلدهای فرم، از عبارت‌های HTML و یا یک اسکریپت استفاده کرده باشد. در ASP.Net + MVC از مکانیزم ValidationRequest برای مقابله با چنین حملاتی ( XSS ) استفاده شده است.
یک فرم با یک Textbox در یک صفحه قرار دهید و درون Textbox یک تگ HTML بنویسید و آنرا به سرور ارسال کنید، در حالت پیش فرض اتفاقی که خواهد افتاد اینست که با یک صفحه‌ی خطا روبرو خواهید شد که به شما هشدار می‌دهد داده‌های ارسال شده، دارای مقادیر غیرمجازی می‌باشند. شما می‌توانید این مکانیزم اعتبارسنجی-امنیتی را غیرفعال کنید. برای غیرفعال کردن آن هم در لینکی که در فوق معرفی گردید توضیحات کافی داده شده است. ولی توصیه میشود برای جلوگیری از حملات XSS هیچگاه این مکانیزم را غیرفعال نکنید، مگر اینکه هیچ داده‌ای را بدون Encode کردن در صفحه نمایش ندهید.
راه‌های زیادی برای هندل کردن این خطا و مطلع کردن کاربر وجود دارند و معمولا از صفحه‌ی خطای سفارشی برای اینکار استفاده میشود. اما ایده‌ی بهتری نیز برای مقابله با این خطا وجود دارد!
فرض کنید اطلاعات فرم شما از طریق Ajax به سرور ارسال می‌شود و نتیجه بصورت Json به مروگر برگشت داده می‌شود. در حالت معمول در صورت رخ دادن خطای فوق، سرور کد 500 را برگشت می‌دهد و تنها راه هندل کردن آن در رویداد error شئ Ajax شما می‌باشد؛ آن‌هم با یک پیغام ساده. ولی من ترجیح می‌دهم به جای صادر کردن خطای 500، در صورت وقوع این خطا آن‌را بصورت یک خطای ModelState به کاربر نمایش دهم. به نظر من این‌کار وجهه‌ی بهتری دارد و ضمنا لازم نیست در هر رویدادی این خطا را هندل کنید. برای رسیدن به این هدف هم چندین راه وجود دارد که ساده‌ترین آن‌ها اینست که یک ModelBinder سفارشی ایجاد کنید و با هندل کردن این خطا، یک پیام خطا را به ModelState اضافه کنید و بعد به MVC بگویید که از این ModelBinder برای پروژه‌ی من استفاده کن. با این روش هرگاه کاربر داده‌ای حاوی کدهای HTML درون فیلدهای فرم وارد کرده باشد، بدون هیچ کار اضافه‌ای وضعیت ModelState شما Invalid می‌شود و خطای مدل به کاربر نمایش داده خواهد شد.
ابتدا کلاسی برای ModelBinder سفارشی خود ایجاد کنید و از کلاس DefaultModelBinder ارث بری کنید. سپس با بازنویسی متد BindModel آن منطق خود را پیاده سازی کنید:

```
using System.Web;
using System.Web.Mvc;
using System.Web.Helpers;
using System.Globalization;

namespace Parsnet
{
    public class ParsnetModelBinder : DefaultModelBinder
    {
        public override object BindModel(ControllerContext controllerContext, ModelBindingContext
bindingContext)
        {
            try
            {
                return base.BindModel(controllerContext, bindingContext);
            }
            catch (HttpRequestValidationException ex)
            {
                var modelState = new ModelState();
                modelState.Errors.Add("اطلاعات ارسالی شما دارای کدهای HTML می‌باشند.");
                var key = bindingContext.ModelName;
                var value =
controllerContext.RequestContext.HttpContext.Request.Unvalidated().Form[key];
                modelState.Value = new ValueProviderResult(value, value, CultureInfo.InvariantCulture);
                bindingContext.ModelState.Add(key, modelState);
            }

            return null;
        }
    }
}
```

در این کلاس ابتدا سعی میکنیم بطور عادی کار را به متد BindModel کلاس پایه بسپاریم. اگر داده‌های ارسال شده حاوی کد HTML

نباشد، بدون هیچ خطایی Model Binding صورت می‌گیرد. ولی در صورتیکه کاربر در فیلدی، از کدهای HTML استفاده کرده باشد، یک خطای HttpRequestValidationException رخ خواهد داد که با هندل کردن آن هدف خود را تامین می‌کنیم.
برای اینکار در قسمت catch بلوک مدیریت خطا، ابتدا یک نمونه از کلاس ModelState را میسازیم. بعد پیام خطای مورد نظر خود را به Errorsهای آن اضافه میکنیم. حال باید یک زوج کلید/مقدار برای این ModelState تعریف کنیم و به bindingContext اضافه کنیم. کلید ما در اینجا نام مدل جاری و مقدار آن هم نام فیلدی از فرم است که سبب بروز این خطا شده است.
حالا نهایت کاری که باید انجام دهیم اینست که در رویداد Application_Start این مدل بایندیگ سفارشی را جایگزین مدل بایندیگ پیش فرض کنیم:

```
ModelBinders.Binders.DefaultBinder = new ParsnetModelBinder();
```

کار تمام است. از حالا به بعد در صورتیکه کاربر در فیلدهای هر فرمی از سایت شما از کدهای HTML استفاده کند دیگر خطایی رخ نمی‌دهد و فقط ModelState در وضعیت Invalid قرار میگیرد که میتوانید با قرار دادن یک ValidationMessage یا ValidationSummary به راحتی خطا را به کاربر نشان دهید.