

این مطلب در ادامه بحث « [اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های ASP.NET MVC](#) » می‌باشد. بنابراین تعاریف مدل و کنترلر آن، به همراه توضیحات ذکر شده در آن، در ادامه مورد استفاده قرار خواهند گرفت.

اصول نمایش Popover در Twitter bootstrap

Popoverها نیز یکی دیگر از کامپوننت‌های جاوا اسکریپتی مجموعه بوت استرپ هستند. بسیار شبیه به Tooltip بوده، اما ماندگارتر هستند. Popoverها با کلیک بر روی یک عنصر باز شده و تنها با کلیک مجدد بر روی آن المان، بسته می‌شوند (البته این موارد نیز قابل تنظیم هستند).

```
<a rel="popover"
    data-content="محتوایی برای نمایش"
    data-original-title="عنوان" href="#">اطلاعات</a>

<script type="text/javascript">
    $(document).ready(function () {
        $('[rel="popover"]').popover({ placement: 'left' })
        .click(function (e) { e.preventDefault(); });
    });
</script>
```

نحوه استفاده از آن را در مثال فوق مشاهده می‌کنید. در اینجا یک لینک با rel=popover تعریف شده است. از این rel، در یافتن کلیه المان‌هایی اینگونه، توسط jQuery استفاده خواهیم کرد. سپس مقدار ویژگی data-content، محتوای اطلاعاتی را که باید نمایش داده شود، مشخص می‌کند. همچنین برای مشخص ساختن عنوان آن می‌توان از ویژگی data-original-title استفاده کرد. نهایتاً نیاز است افزونه popover بر روی المان‌هایی با rel=popover فراخوانی گردد. در روال رخدادگردان click آن، با استفاده از e.preventDefault، سبب خواهیم شد تا با کلیک بر روی لینک تعریف شده، صفحه مجدداً بازیابی نشده و مکان اسکرول عمودی صفحه، تغییر نکند.



تبدیل خطاهای اعتبارسنجی ASP.NET MVC به PopOver

هدف ما در اینجا نهایتاً رسیدن به شکل زیر می‌باشد:

تعریف کاربر جدید

خطای اعتبارسنجی

لطفا نام را تکمیل کنید

نام

نام خانوادگی

اطلاعات

ارسال

همانطور که ملاحظه می‌کنید، اینبار بجای نمایش خطاها در یک برچسب، مقابل کنترل متناظر، این خطا صرفاً در حالت فوکوس کنترل، به شکل یک PopOver در کنار آن ظاهر شده است.

کدهای کامل View برنامه

```
@model Mvc4TwitterBootstrapTest.Models.User
@{
    ViewBag.Title = "Index";
}
@using (Html.BeginForm())
{
    @Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })

    <fieldset class="form-horizontal">
        <legend>تعریف کاربر جدید</legend>
        <div class="control-group">
            @Html.LabelFor(model => model.Name, new { @class = "control-label" })
            <div class="controls">
                @Html.EditorFor(model => model.Name)
                *@Html.ValidationMessageFor(model => model.Name, null, new { @class = "help-inline"
            })*@
            </div>
        </div>
        <div class="control-group">
            @Html.LabelFor(model => model.LastName, new { @class = "control-label" })
            <div class="controls">
                @Html.EditorFor(model => model.LastName)
                *@Html.ValidationMessageFor(model => model.LastName, null, new { @class = "help-
inline" })*@
            </div>
        </div>
        <div class="form-actions">
            <button type="submit" class="btn btn-primary">
                ارسال</button>
            </div>
    </fieldset>
}
@section JavaScript
{
    <script type="text/javascript">
        $.validator.setDefaults({
            showErrors: function (errorMap, errorList) {
                this.defaultShowErrors();
                //اگر المانی معتبر است نیاز به نمایش پاپ ندارد
                $(". " + this.settings.validClass).popover("destroy");
                //افزودن پاپ اورها
                for (var i = 0; i < errorList.length; i++) {
                    var error = errorList[i];
```

```

        $(error.element).popover({ placement: 'left' })
        .attr("data-original-title", "خطای اعتبارسنجی")
        .attr("data-content", error.message);
    }
},
// همانند قبل برای رنگی کردن کل ردیف در صورت عدم اعتبار سنجی و برعکس
highlight: function (element, errorClass, validClass) {
    if (element.type === 'radio') {
        this.findByName(element.name).addClass(errorClass).removeClass(validClass);
    } else {
        $(element).addClass(errorClass).removeClass(validClass);
        $(element).closest('.control-group').removeClass('success').addClass('error');
    }
    $(element).trigger('highlighted');
},
unhighlight: function (element, errorClass, validClass) {
    if (element.type === 'radio') {
        this.findByName(element.name).removeClass(errorClass).addClass(validClass);
    } else {
        $(element).removeClass(errorClass).addClass(validClass);
        $(element).closest('.control-group').removeClass('error').addClass('success');
    }
    $(element).trigger('unhighlighted');
}
});
// برای حالت پست بک از سرور عمل می‌کند
$(function () {
    $('form').each(function () {
        $(this).find('div.control-group').each(function () {
            if ($(this).find('span.field-validation-error').length > 0) {
                $(this).addClass('error');
            }
        });
    });
});
</script>
}

```

کدهای مدل و کنترلر، همانند مطلب « [اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های ASP.NET MVC](#) » می‌باشند و از تکرار مجدد آن‌ها در اینجا صرف‌نظر گردید.

توضیحات

- با توجه به اینکه دیگر نمی‌خواهیم خطاها به صورت برجسته در مقابل کنترل‌ها نمایش داده شوند، کلیه `Html.ValidationMessageFor` به صورت کامنت درآورده شده‌اند.

- تغییر دوم مطلب جاری، اضافه شدن متد `showErrors` به تنظیمات پیش فرض `jQuery Validator` است. در این متد، اگر المانی معتبر بود، Popover آن حذف می‌شود یا در سایر حالات، المان‌هایی که نیاز به اعتبارسنجی سمت کلاینت دارند، یافت شده و سپس ویژگی `data-content` با مقداری معادل خطای اعتبارسنجی متناظر، به این المان افزوده و سپس متد `popover` بوت استرپ بر روی آن فراخوانی می‌گردد.

به عبارتی زمانیکه یک `input box` در `ASP.NET MVC` به همراه مقادیر مرتبط با اعتبارسنجی آن رندر می‌شود، چنین شکلی را خواهد داشت:

```

<input class="text-box single-line" data-val="true" data-val-required="لطفا نام را تکمیل کنید"
id="Name" name="Name" type="text" value="" />

```

اما در اینجا به صورت پویا، `data-content` و `data-original-title` نیز به آن افزوده می‌گردند:

```

<input class="text-box single-line input-validation-error" data-val="true" data-val-required="لطفا نام را تکمیل کنید"
id="Name" name="Name" type="text" value=""
data-original-title="خطای اعتبارسنجی" title="" data-content="لطفا نام را تکمیل کنید">

```

این مقادیر توسط افزونه `popover` بوت استرپ شناسایی شده و مورد استفاده قرار می‌گیرد. البته این موارد را در صورت نیاز به صورت دستی نیز می‌توان تعریف و اضافه کرد:

```
@Html.TextBoxFor(x => x.Name,
    new { data_content = "Name is required",
          data_original_title = "Error", rel="popover" })
```

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۵ ۱۴:۴۱

دریافت مثال این قسمت: [BootstrapPopover.zip](#)

+ [نحوه بازسازی وابستگی‌ها](#)