

Twitter Bootstrap یک فریم ورک CSS بسیار محبوب سورس باز تولید برنامه‌های وب به کمک CSS، HTML و جاوا اسکریپت است. این فریم ورک حاوی بسیاری از المان‌های مورد نیاز جهت تولید وب سایت‌هایی زیبا، مانند دکمه‌ها، عناصر فرم‌ها، منوها، ویجت‌ها و غیره است. تمام این‌ها نیز همانطور که عنوان شد بر مبنای CSS، HTML و جاوا اسکریپت تهیه شده‌اند؛ بنابراین در هر نوع فناوری سمت سروری مانند PHP، ASP.NET، روبی و امثال آن قابل استفاده است.

## دریافت Twitter Bootstrap

محل اصلی دریافت Twitter Bootstrap، آدرس ذیل است:

<http://twitter.github.com/bootstrap>

البته ما از آن در اینجا به شکل خام فوق استفاده نخواهیم کرد؛ زیرا نیاز است قابلیت‌های استفاده در محیط‌های راست به چپ فارسی نیز به آن اضافه شود. برای این منظور می‌توانید از یکی از دو بسته نیوگت ذیل استفاده نمایید:

Twitter.BootstrapRTL, <https://nuget.org/packages/Twitter.BootstrapRTL>

RTL Twitter Bootstrap, <https://nuget.org/packages/Twitter.Bootstrap.RTL>

و یا حتی از منابع سایت <http://rbootstrap.ir> نیز می‌توان استفاده کرد. برای نمونه دستور زیر را در کنسول پاورشل ویژوال استودیو وارد نمایید تا اسکریپت‌ها و فایل‌های CSS مورد نیاز به پروژه جاری اضافه شوند:

```
PM> Install-Package Twitter.BootstrapRTL
```

پس از نصب، شاهد افزوده شدن چنین فایل‌هایی به یک پروژه ASP.NET MVC خواهیم بود:



در اینجا فایل‌های min، نگارش‌های فشرده شده فایل‌های js یا css هستند که با توجه به امکانات اضافه شده به ASP.NET MVC4، از آن‌ها استفاده نخواهیم کرد و برای افزودن و تعریف آن‌ها از امکانات Bundling and minification توکار فریم ورک ASP.NET MVC به نحوی که در ادامه توضیح داده خواهد شد، استفاده می‌کنیم.

فایل‌های png اضافه شده، آیکون‌های مخصوص Twitter Bootstrap هستند که اصطلاحاً به آن‌ها sprite images نیز گفته می‌شود. در این نوع تصاویر، تعداد زیادی آیکون در کنار هم، برای بهینه سازی تعداد بار رفت و برگشت به سرور جهت دریافت تصاویر، طراحی شده و قرار گرفته‌اند.

فایل‌های js این مجموعه اختیاری بوده و برای استفاده از ویجت‌های Twitter Bootstrap مانند آکاردئون کاربرد دارند. این فایل‌ها برای اجرا، نیاز به jQuery خواهند داشت.

### افزودن تعاریف اولیه Twitter Bootstrap به یک پروژه ASP.NET MVC

امکانات Bundling and minification در نوع پروژه‌های نسبتاً جامع‌تر ASP.NET MVC به صورت پیش فرض لحاظ شده است. اما اگر یک پروژه خالی را شروع کرده‌اید، نیاز است بسته نیوگت آن‌را نیز نصب کنید:

```
PM> Install-Package Microsoft.AspNet.Web.Optimization
```

پس از آن، کلاس کمکی BundleConfig ذیل را به پروژه جاری اضافه نمایید. به کمک آن قصد داریم امکانات Bundling and minification را فعال کرده و همچنین به آن بگوییم اسکریپت‌ها و فایل‌های CSS تعریف شده را به همان نحوی که معرفی می‌کنیم، پردازش کن (توسط کلاس سفارشی AsIsBundleOrderer).

```
using System.Collections.Generic;
using System.IO;
```

```

using System.Web;
using System.Web.Optimization;

namespace Mvc4TwitterBootstrapTest.Helper
{
    /// <summary>
    /// A custom bundle orderer (IBundleOrderer) that will ensure bundles are
    /// included in the order you register them.
    /// </summary>
    public class AsIsBundleOrderer : IBundleOrderer
    {
        public IEnumerable<FileInfo> OrderFiles(BundleContext context, IEnumerable<FileInfo> files)
        {
            return files;
        }
    }

    public static class BundleConfig
    {
        private static void addBundle(string virtualPath, bool isCss, params string[] files)
        {
            BundleTable.EnableOptimizations = true;

            var existing = BundleTable.Bundles.GetBundleFor(virtualPath);
            if (existing != null)
                return;

            Bundle newBundle;
            if (HttpContext.Current.IsDebuggingEnabled)
            {
                newBundle = new Bundle(virtualPath);
            }
            else
            {
                newBundle = isCss ? new Bundle(virtualPath, new CssMinify()) : new Bundle(virtualPath,
new JsMinify());
            }
            newBundle.Orderer = new AsIsBundleOrderer();

            foreach (var file in files)
                newBundle.Include(file);

            BundleTable.Bundles.Add(newBundle);
        }

        public static IHtmlString AddScripts(string virtualPath, params string[] files)
        {
            addBundle(virtualPath, false, files);
            return Scripts.Render(virtualPath);
        }

        public static IHtmlString AddStyles(string virtualPath, params string[] files)
        {
            addBundle(virtualPath, true, files);
            return Styles.Render(virtualPath);
        }

        public static IHtmlString AddScriptUrl(string virtualPath, params string[] files)
        {
            addBundle(virtualPath, false, files);
            return Scripts.Url(virtualPath);
        }

        public static IHtmlString AddStyleUrl(string virtualPath, params string[] files)
        {
            addBundle(virtualPath, true, files);
            return Styles.Url(virtualPath);
        }
    }
}

```

نکته دیگری که در این کلاس سفارشی در نظر گرفته شده، عدم فعال سازی مباحث Bundling and minification در حالت Debug است. اگر در وب کانفیگ، تنظیمات پروژه را بر روی Release قرار دهید فعال خواهد شد (مناسب برای توزیع) و در حالت توسعه برنامه (حالت دیباگ)، این اسکریپت‌ها و فایل‌های CSS، قابلیت دیباگ و بررسی را نیز خواهند داشت.

پس از افزودن کلاس‌های کمکی فوق، به فایل layout پروژه مراجعه کرده و تعاریف ذیل را به ابتدای فایل اضافه نمایید:

```
@using Mvc4TwitterBootstrapTest.Helper
<!DOCTYPE html>
<html>
<head>
<title>@ViewBag.Title</title>
@BundleConfig.AddStyles("~/Content/css",
    "~/Content/bootstrap.css",
    "~/Content/bootstrap-responsive.css",
    "~/Content/Site.css"
)
@BundleConfig.AddScripts("~/Scripts/js",
    "~/Scripts/jquery-1.9.1.min.js",
    "~/Scripts/jquery.validate.min.js",
    "~/Scripts/jquery.unobtrusive-ajax.min.js",
    "~/Scripts/jquery.validate.unobtrusive.min.js",
    "~/Scripts/bootstrap.min.js"
)
@RenderSection("JavaScript", required: false)
</head>
```

و اگر برنامه را اجرا کنید، بجای حداقل 8 مدخل اشاره کننده به فایل های اسکریپت و CSS مورد نیاز یک برنامه ASP.NET MVC، فقط دو مدخل ذیل را مشاهده خواهید نمود:

```
<!DOCTYPE html>
<html>
<head>
<title>Index</title>
<link href="/Content/css?v=vsUQD00Jg4AJ-RZH8jSRRCu_rj12U1nZrmSsaUyxAc1" rel="stylesheet"/>
<script src="/Scripts/js?v=GezdoTDiWY3acc3mI2Ujm_7nKKzh6Lu1Wr8TGyyLpW41"></script>
</head>
```

به این ترتیب تعداد رفت و برگشت های مرورگر به سرور، برای دریافت فایل های مورد نیاز جهت رندر صحیح یک صفحه، به شدت کاهش خواهد یافت. همچنین این فایل ها در حالت release فشرده نیز خواهند شد؛ به همراه تنظیم خودکار هدر کش شدن آن ها برای مدتی مشخص، جهت کاهش بار سرور و کاهش پهنای باند مصرفی آن.

## مفاهیم پایه ای Twitter Bootstrap

### الف) Semantic class names

به عبارتی کلاس های Twitter Bootstrap دارای نام هایی معنا دار و مفهومی می باشند؛ مانند کلاس های CSS ای، به نام های Success، Error، Info و امثال آن. این نام ها مفهومی را می رسانند؛ اما در مورد نحوه پیاده سازی آن ها جزئیاتی را بیان نمی کنند. برای نمونه می توان کلاسی را به نام redText ایجاد کرد. هر چند این نام، توضیحاتی را در مورد علت وجودی اش بیان می کند، اما بسیار ویژه بوده و در مورد جزئیات پیاده سازی آن نیز اطلاعاتی را ارائه می دهد. در این حالت redText معنایی ندارد. چرا یک Text باید قرمز باشد؟ برای مثال این متن قرمز است چون مثلاً شخصی، به آن رنگ ویژه علاقه دارد، یا اینکه قرمز است بخاطر نمایش خطایی در صفحه؟ به همین جهت در Twitter Bootstrap از نام های مفهومی یاده شده، مانند Error استفاده می شود. نام هایی معنا دار اما بدون دقیق شدن در مورد ریز جزئیات پیاده سازی آن ها. در این حالت می توان [قالب جدیدی را تدارک دید](#) و با ارائه تعاریف جدیدی برای کلاس Error و نحوه نمایش دلخواهی را به آن اعمال نمود. یا برای نمونه نام rightside را برای نمایش ستونی در صفحه، در نظر بگیرید. این نام بسیار ویژه است؛ اما Sematic name آن می تواند sidebar باشد تا بدون دقیق شدن در جزئیات پیاده سازی آن، در چپ یا راست صفحه قابل اعمال باشد. Semantic class names کلید هایی هستند جهت استفاده مجدد از قابلیت های یک فریم ورک CSS.

### ب) Compositional classes

اکثر کلاس های Twitter Bootstrap دارای محدوده کاری کوچکی هستند و به سادگی قابل ترکیب با یکدیگر جهت رسیدن به نمایی خاص می باشند. برای مثال به سادگی می توان به یک table سه ویژگی color، hover و width برگرفته شده از Twitter Bootstrap را انتساب داد و نهایتاً به نتیجه دلخواه رسید؛ بدون اینکه نگران باشیم افزودن کلاس جدیدی در اینجا بر روی سایر کلاس های انتساب داده شده، تاثیر منفی دارد.

### Conventions (ج)

برای استفاده از اکثر قابلیت‌های این فریم ورک CSS یک سری قراردادهای پیش فرضی وجود دارند. برای مثال اگر از کلاس توکار pagination به همراه یک سری ul و li استفاده کنید، به صورت خودکار یک pager شکل ظاهر خواهد شد. یا برای مثال اگر به یک html table کلاس‌های table-hover table-striped table را انتساب دهیم، به صورت خودکار قراردادهای پیش فرض table مجموعه Twitter Bootstrap به آن اعمال شده، به همراه رنگی ساختن یک درمیان زمینه ردیف‌ها و همچنین فعال سازی تغییر رنگ ردیف‌ها با حرکت ماوس از روی آن‌ها.

### طرح‌بندی صفحات یک سایت به کمک Twitter Bootstrap

#### بررسی Grid layouts

Layout به معنای طرح‌بندی و چیدمان محتوا در یک صفحه است. یکی از متداولترین روش‌های طرح‌بندی صفحات چه در حالت چاپی و چه در صفحات وب، چیدمان مبتنی بر جداول و گریدها است. از این جهت که نحوه سیلان و نمایش محتوا از چپ به راست و یا راست به چپ را به سادگی میسر می‌سازند؛ به همراه اعمال حاشیه‌های مناسب جهت قسمت‌های متفاوت محتوای ارائه شده. Grid در طرح‌بندی، نمایش بصری نخواهد داشت اما در ساختار صفحه وجود داشته و مباحثی مانند جهت، موقعیت و یکپارچگی و یکدستی طراحی را سبب می‌شود. به علاوه مرورگرها و مفهوم Grid نیز به خوبی با یکدیگر سازگار هستند. در دنیای HTML و CSS، طراحی‌ها بر اساس مفهوم ساختار مستطیلی اشیاء صورت می‌گیرد:



برای نمونه در اینجا تصویر [CSS Box Model](#) را مشاهده می‌کنید. به این ترتیب، هر المان دارای محدوده‌ای مستطیلی با طول و عرض مشخص، به همراه ویژگی‌هایی مانند Margin، Border و Padding است. در سال‌های اولیه طراحی وب، عموماً کارهای طراحی صفحات به کمک HTML Tables انجام می‌شد. اما با پخته‌تر شدن CSS، استفاده از Tables برای طراحی صفحات کمتر و کمتر گشت تا اینکه نهایتاً فریم ورک‌های CSS ایی پدید آمدند تا طراحی‌های مبتنی بر CSS را با ارائه گریدها، ساده‌تر کنند. مانند 960 GS، Blue print، و ... Twitter Bootstrap که طراحی مبتنی بر گریدهای CSS ایی را به مجموعه قابلیت‌های دیگر خود افزوده است.

#### بررسی Fixed Grids

در اینجا در صفحه layout برنامه، یک Div دربرگیرنده دو Div دیگر را مشاهده می‌کنید:

```
<body>
  <div>
    <div>
      <h1>
        Title Title
      </h1>
      Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text
    </div>
    <div>
      @RenderBody()
    </div>
  </div>
</body>
```

اگر در این حالت صفحه را در مرورگر بررسی کنیم، تمام قسمت‌های نمایش داده شده به همین نحو از بالا به پایین صفحه قرار خواهند گرفت. اما قصد داریم این محتوا را، دو ستونی نمایش دهیم. Div به همراه Title در یک طرف صفحه و Div دربرگیرنده محتوای صفحات، در قسمتی دیگر.

برای اینکار در Twitter Bootstrap از کلاسی به نام row استفاده می‌شود که بیانگر یک ردیف است. این کلاس را به خارجی‌ترین Div موجود اعمال خواهیم کرد. در یک صفحه، هر تعداد row ایی را که نیاز باشد، می‌توان تعریف کرد. داخل این ردیف‌ها، امکان تعریف ستون‌های مختلف و حتی تعریف ردیف‌های تو در تو نیز وجود دارد. هر ردیف Twitter Bootstrap از 12 ستون تشکیل می‌شود و برای تعریف آن‌ها از کلاس span استفاده می‌گردد. در این حالت جمع اعداد ذکر شده پس از 12 باید 12 را تشکیل دهند.

```
<body>
  <div class="row">
    <div class="span7">
      <h1>
        Title Title
      </h1>
      Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text
    </div>
    <div class="span5">
      @RenderBody()
    </div>
  </div>
</body>
```

برای نمونه در اینجا کلاس row به خارجی‌ترین Div موجود اعمال شده است. دو Div داخلی هر کدام دارای کلاس‌های span ایی جهت تشکیل ستون‌های داخل این ردیف هستند. جمع اعداد بکارگرفته شده پس از آن‌ها، عدد 12 را تشکیل می‌دهد. به این ترتیب به یک طراحی دو ستونی خواهیم رسید.



در این تصویر، قسمت `RenderBody` کار رندر اکشن متد `Index` کنترلر `Home` برنامه را با `View`ی معادل کدهای ذیل، انجام داده است:

```
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
<div class="hero-unit">
    <h2>@ViewBag.Message</h2>
    <p>
        This is a template to demonstrate the way to beautify the default MVC template
        using Twitter Bootstrap website. It is pretty simple and easy.</p>
    <p>
        <a href="http://asp.net/mvc" class="btn btn-primary btn-large" style="color: White;">
            To learn more about ASP.NET MVC visit &raquo;</a></p>
    </div>
```

### درک نحوه عملکرد Grid در Twitter Bootstrap

در مثال ذیل 5 ردیف را مشاهده می‌کنید:

```
<div class="row">
```

```

<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
</div>
<div class="row">
  <div class="span3">3</div>
  <div class="span4">4</div>
  <div class="span5">5</div>
</div>
<div class="row">
  <div class="span5">5</div>
  <div class="span7">7</div>
</div>
<div class="row">
  <div class="span3">3</div>
  <div class="span7 offset2">7 offset 2</div>
</div>
<div class="row">
  <div class="span12">12</div>
</div>

```

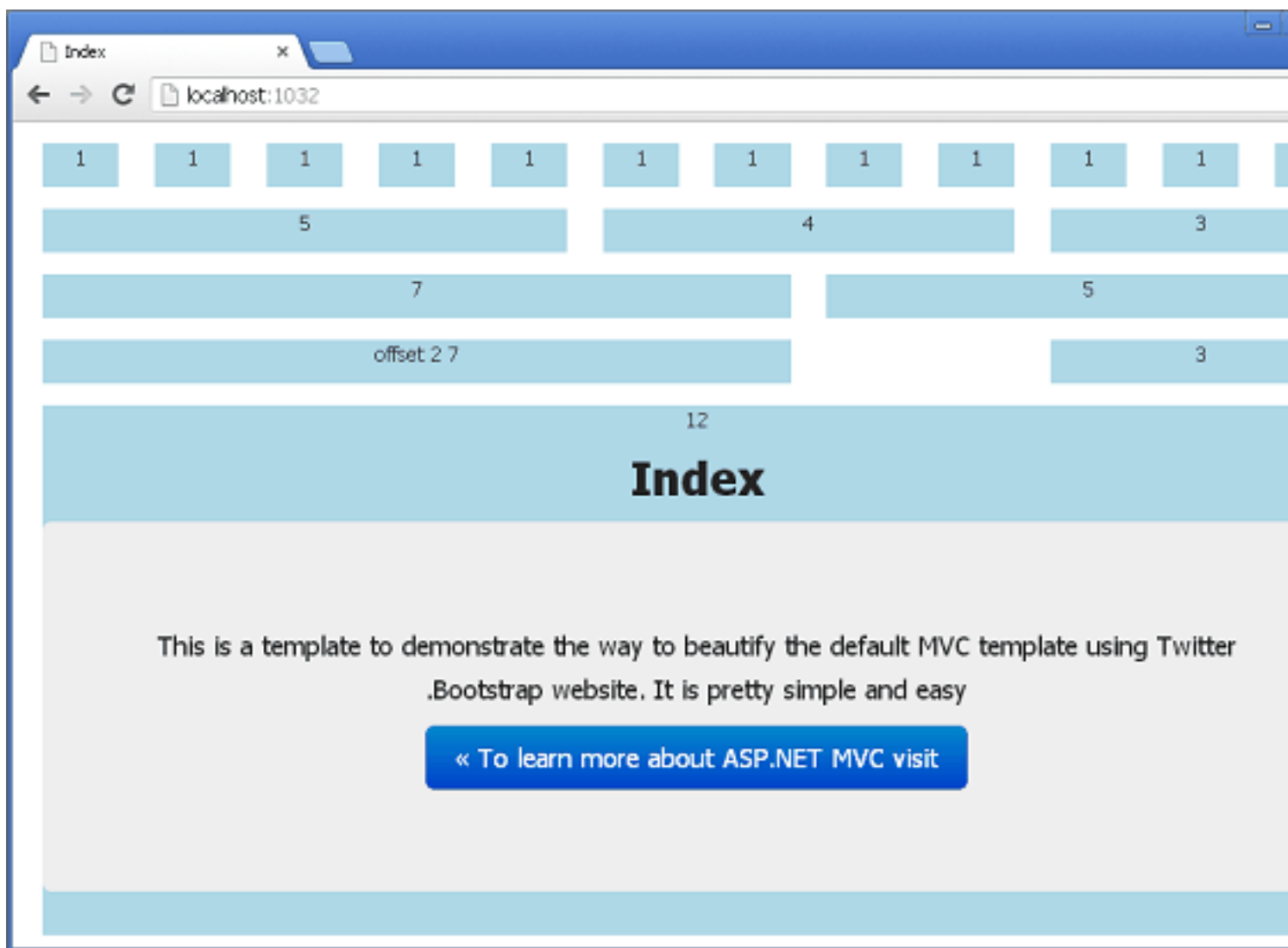
در هر ردیف، امکان استفاده از حداکثر 12 ستون وجود دارد که یا می‌توان مانند ردیف اول، 12 ستون را ایجاد کرد و یا مانند ردیف دوم، سه ستون با عرض‌های متفاوت (و یا هر ترکیب دیگری که به جمع 12 برسد). در ستون چهارم، از کلاس offset نیز استفاده شده است. این مورد سبب می‌شود ستون جاری به تعدادی که مشخص شده است به سمت چپ (با توجه به استفاده از حالت RTL در اینجا) رانده شود و سپس ترسیم گردد. یا اینکه می‌توان مانند ردیف آخر، یک ستون را به عرض 12 که در حقیقت 940 پیکسل است، ترسیم نمود. برای اینکه بتوانیم این گرید تشکیل شده و همچنین ستون‌ها را بهتر مشاهده کنیم، به فایل style.css سایت، تنظیم زیر را اضافه کنید:

```

[class*="span"]
{
background-color: lightblue;
text-align: center;
margin-top: 15px;
}

```





نکته جالب این گرید، Responsive یا واکنشگرا بودن آن است. در این حالت، عرض مرورگر را کم و زیاد کنید. خواهید دید که ستون‌ها در صورتیکه در عرض نمایشی جاری، قابل ارائه نباشند، به ردیف‌های بعدی منتقل خواهند شد. البته باید دقت داشت که این گرید هیچگاه یک ستون را نخواهد شکست. برای نمونه ردیف آخر، همواره با همان عرض ثابتش نمایش داده می‌شود و با کوچکتر کردن اندازه مرورگر، یک اسکرول افقی برای نمایش محتوای آن ظاهر خواهد شد.

#### یک نکته

اگر نمی‌خواهید که چنین رفتار واکنش‌گرایی بروز کند، نیاز است کلیه ردیف‌ها را در div ایی با کلاسی به نام container محصور کنید.

به این ترتیب ابتدا گرید نمایش داده شده، در میانه صفحه ظاهر خواهد شد (پیشتر از سمت راست شروع شده بود). همچنین دیگر با کوچک و بزرگ شدن اندازه مرورگر، ستون‌ها به شکل یک پشته بر روی هم قرار نخواهند گرفت. (اگر پس از این تنظیم، چنین قابلیتی را مشاهده نکردید و هنوز هم طراحی، واکنشگرا بود، تعریف bootstrap-responsive.css را نیاز است برای آزمایش، از هدر صفحه حذف کنید)

#### بررسی Fluid Grids

به گرید قسمت قبل از این جهت Fixed Grid گفته می‌شود که عرض هر span آن با یک عدد مشخص تعیین گشته است. اما در

حالت Fluid Grid، عرض هر span برحسب درصد تعیین می‌شود. بکارگیری درصد در اینجا به معنای امکان تغییر عرض یک ستون بر اساس عرض جاری Container آن است. در اینجا span12 دارای عرض 100 درصد خواهد بود. در مثال قبل، برای استفاده از Fluid grids، تنها کافی است هرجایی کلاسی مساوی row وجود دارد، به row-fluid تغییر کند. همچنین کلاس container را به container-fluid تغییر دهید. برای آزمایش آن، اندازه و عرض نمایشی مرورگر خود را تغییر دهید. اینبار مشاهده خواهید کرد که برخلاف حالت Fixed Grid، عرض ستون‌ها به صورت خودکار کم و زیاد می‌شوند. این مورد بر روی محتوای قرار گرفته در این ستون‌ها نیز تاثیر گذار است. برای مثال اگر یک تصویر را در حالت Fluid grid در ستونی قرار دهید، با تغییر عرض مرورگر، اندازه این تصویر نیز تغییر خواهند کرد؛ اما در حالت Fixed Grid خیر. حالت Fluid، شیوه متداول استفاده از bootstrap در اکثر سایت‌های مهمی است که تابحال از این فریم ورک CSS استفاده کرده‌اند.



## مروری بر طراحی واکنشگرا یا Responsive

این روزها تعدادی از کاربران، با استفاده از ابزارهای موبایل و تبلت‌ها از وب سایت‌ها بازدید می‌کنند. هر کدام از این‌ها نیز دارای اندازه نمایشی متفاوتی می‌باشند. بنابراین نیاز خواهد بود تا حالت بهینه‌ای را جهت اینگونه وسایل نیز طراحی نمود. حالت بهینه در اینجا به معنای قابل خواندن بودن متون، امکان استفاده از لینک‌های ورود به صفحات مختلف و همچنین حذف اسکروول و مباحث زوم جهت مشاهده صفحات است.

یکی از ویژگی‌های CSS به نام media چنین قابلیتی را فراهم می‌سازد. برای نمونه قسمتی از فایل bootstrap-responsive.css دارای چنین تعاریفی است:

```
@media (min-width: 768px) and (max-width: 979px) {
  .hidden-desktop {
    display: inherit !important;
  }
}
```

توسط این تنظیمات، شیوه نامه تعیین شده، تنها به صفحاتی با اندازه‌هایی بین 768 تا 979 پیکسل اعمال می‌شود (تعدادی از تبلت‌ها برای نمونه).

Bootstrap برای مدیریت اندازه‌های مختلف وسایل موبایل، شیوه‌نامه‌های خاصی را تدارک دیده است که از اندازه 480px و یا کمتر، تا 1200px و یا بیشتر را پوشش می‌دهد.

به این ترتیب با اندازه 940px که پیشتر در مورد آن بحث شد، اندازه span12 به صورت خودکار به اندازه‌های متناسب با صفحات نمایشی کوچکتر تنظیم می‌گردد. همچنین برای اندازه‌های صفحات کوچکتر از 768px به صورت خودکار از Fluid columns استفاده می‌گردد.

تنها کاری که برای اعمال این قابلیت باید صورت گیرد، افزودن تعاریف فایل bootstrap-responsive.css به هدر صفحه است که در قسمت قبل انجام گردید. این فایل باید پس از فایل اصلی bootstrap.css اضافه شود.

## کلاس‌های کمکی طراحی واکنشگرا

Bootstrap شامل تعدادی کلاس کمکی در فایل bootstrap-responsive.css خود می‌باشد شامل visible-phone، visible-tablet و visible-desktop به علاوه hidden-phone، hidden-tablet و hidden-desktop. به این ترتیب می‌توان محتوای خاصی را جهت وسایل ویژه‌ای نمایان یا مخفی ساخت.

برای مثال محتوای مشخص شده با کلاس hidden-desktop، در اندازه وسایل دسکتاپ نمایش داده نخواهد شد.

برای آزمایش آن، شش div را با کلاس‌های یاد شده و محتوایی دلخواه ایجاد کرده و سپس اندازه عرض مرورگر را تغییر دهید تا بهتر بتوان مخفی یا نمایان ساختن محتوا را بر اساس اندازه صفحه جاری درک کرد.

یکی از کاربردهای این قابلیت، قرار دادن تبلیغاتی با اندازه‌های تصاویری مشخص برای وسایل مختلف است؛ بجای اینکه منتظر شویم تا Fluid layout اندازه تصاویر را به صورت خودکار کوچک یا بزرگ کند، که الزاماً بهترین کیفیت را حاصل نخواهد ساخت.

```
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span4">
      <div class="visible-phone">
        visible-phone</div>
      </div>
    <div class="span4">
      <div class="visible-tablet">
        visible-tablet</div>
      </div>
    <div class="span4">
      <div class="visible-desktop">
        visible-desktop</div>
      </div>
    </div>
  </div>
</div>
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span4">
      <div class="hidden-phone">
        hidden-phone</div>
      </div>
    </div>
  </div>
</div>
```

```
<div class="span4">
  <div class="hidden-tablet">
    hidden-tablet</div>
  </div>
  <div class="span4">
    <div class="hidden-desktop">
      hidden-desktop</div>
    </div>
  </div>
</div>
```



## نظرات خوانندگان

نویسنده: احمدعلی شفیعی  
تاریخ: ۱۰:۸ ۱۳۹۲/۰۳/۱۹

محدودیت ۱۲ تایی ستون‌ها به اندازه‌ی صفحه بستگی دارد. برای من ۱۳ تاست.

نویسنده: وحید نصیری  
تاریخ: ۱۷:۳۵ ۱۳۹۲/۰۳/۱۹

شعار این کتابخانه چنین چیزی هست: « [Bootstrap is built on responsive 12-column grids](#) ». یعنی حالت پیش فرض طراحی آن 12 ستونی است. اگر به فایل css آن مراجعه کنید، یک چنین عرض‌هایی قابل مشاهده است:

```
.span12 {
  width: 940px;
}
.span1 {
  width: 60px;
}
```

البته این‌ها برای حالت fixed است و نه fluid که بر اساس درصد تعریف شده

```
.row-fluid .span12 {
  width: 100%;
  *width: 99.94680851063829%;
}
```

container هم با عرض 940 پیکسل تعریف شده. همچنین یک GutterWidth را هم باید در بین ستون‌ها در نظر داشت. حالت پیش فرض:

```
@gridColumns: 12;
@gridColumnWidth: 60px;
@gridGutterWidth: 20px;
```

البته میشه مثلا عرض رو بیشتر از 940 تعریف کرد. برای اینکار باید css رو ویرایش و از ابزارهایی مانند [less استفاده کرد](#) و یا امکان [سفارشی سازی اون](#) هم هست.

نویسنده: وحید نصیری  
تاریخ: ۱۲:۴۸ ۱۳۹۲/۰۳/۲۰

## یک نکته تکمیلی

در مطلب فوق یک چنین تعریفی را مشاهده نمودید:

```
@BundleConfig.AddStyles("~/Content/css",
    "~/Content/bootstrap.css",
    "~/Content/bootstrap-responsive.css",
    "~/Content/Site.css"
)
```

در پوشه content، فایل‌های bootstrap.min.css و bootstrap.css وجود دارند (اگر بسته نیوگت آن‌را نصب کنید). در این بین، اگر فایل bootstrap.css را اندکی ویرایش کنید (مثلا نام فونت‌ها یا اندازه آن‌ها را تغییر دهید)، متوجه خواهید شد که تغییرات شما اصلا اعمال نمی‌شوند. علت چیست؟! سیستم Bundling به صورت هوشمند در صورت وجود فایل‌های css.min، از آن‌ها استفاده خواهد کرد (حتی در حالت دیباگ) و نه

از فایل bootstrap.css ذکر شده در تنظیمات. به همین جهت ممکن است مدتی سر درگم شوید که چرا هر تغییری را که اعمال می‌کنم، تاثیری ندارد!  
بنابراین یا فایل‌های css.min را حذف کنید و یا مکان آن‌ها را تغییر دهید تا در کنار فایل‌های غیرفشرده شده پیش فرض نباشند.

نویسنده: یزدان  
تاریخ: ۱۳۹۲/۰۳/۳۱ ۱:۳۷

آیا برای web form هم کارایی داره و میشه در web form از اون استفاده کرد ؟

```
@BundleConfig.AddStyles("~/Content/css",
    "~/Content/bootstrap.css",
    "~/Content/bootstrap-responsive.css",
    "~/Content/Site.css"
)
```

در وب فرم به چه صورت این رو اضافه کنیم ؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۳/۳۱ ۹:۰۰

در ASP.NET 4.5 مباحث Bundling and Minification [جزو استاندارد آن](#) است. در ASP.NET 4.0 می‌شود از امکانات ASP.NET MVC 4 هم استفاده کرد (همان Bundling and Minification کامپایل شده برای دات نت 4 است).

نویسنده: ایلیا اکبری فرد  
تاریخ: ۱۳۹۲/۰۴/۰۱ ۱۱:۵۰

با سلام.

در قسمت Bundling وقتی از کد شما استفاده میکنم تنها فقط Site.css را باندل میکند و سایر css ها باندل نمی‌شوند. وقتی کد زیر را کامنت کردم مشکل حل شد.

```
@BundleConfig.AddStyles("~/Content/css",
    "~/Content/bootstrap.css",
    "~/Content/Site.css")
...
var existing = BundleTable.Bundles.GetBundleFor(virtualPath);
//if (existing != null)
//    return;
```

کامنت کردن این کد مشکلی ایجاد میکند؟

نویسنده: میثم  
تاریخ: ۱۳۹۲/۰۴/۰۹ ۱۱:۴۰

فایل bootstrap.css بصورت ناقص فشرده می‌شود.  
بطور دقیق‌تر تا رسیدن به کاراکترهای :: بدون مشکل کار پیش می‌رود.  
در حالت اجرا وقتی فایل را css را باز میکنیم فقط کلاس‌ها و شناسه‌های قبل از :: وجود دارند.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۴/۰۹ ۱۱:۵۵

در پروژه MVC خود، خط فرمان پاورشل نیوگت را باز کنید و دستور Update-Package را صادر کنید. تقریباً تمام اجزای MVC4 مرتباً به روز می‌شوند؛ منجمله همین بحث Web.Optimization آن:

PM> Update-Package

نویسنده: سیروان عفیفی  
تاریخ: ۱۹:۲۲ ۱۳۹۲/۰۵/۱۸

دو مدخل برای من اضافه میشن

```
<link href="/Content/myCSS?v=LBa4VrARIJBaPHOXLQoCWVPKu_c6QBb4D7npv-dq5TA1" rel="stylesheet"/>
<script src="/Scripts/js?v=rk-13czPS8KW6usE0pK8aIglVqPOVEKWoSXV4N-5N2Q1"></script>
```

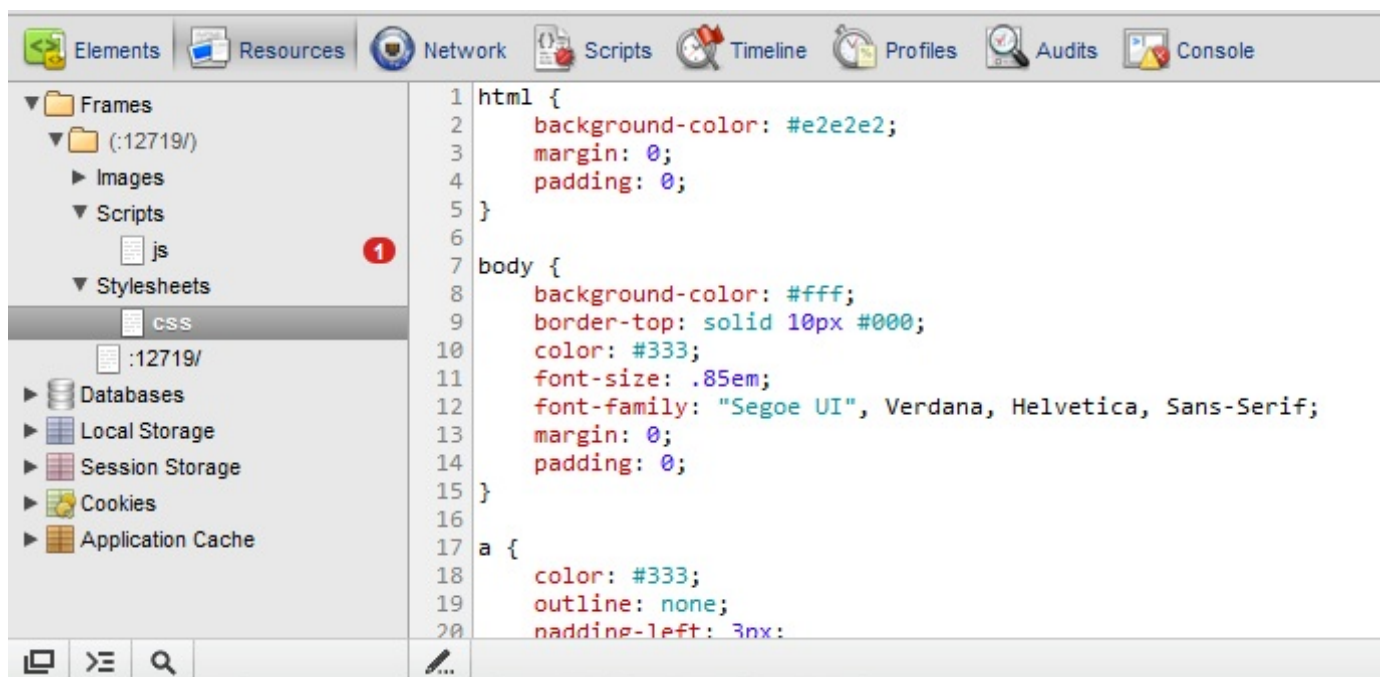
منتها برای CSS فقط فایل Style.css و برای جاوااسکریپت فقط jQuery رو اضافه میکنه، به نظرتون مشکل چی میتونه باشه؟

نویسنده: وحید نصیری  
تاریخ: ۲۱:۷ ۱۳۹۲/۰۵/۱۸

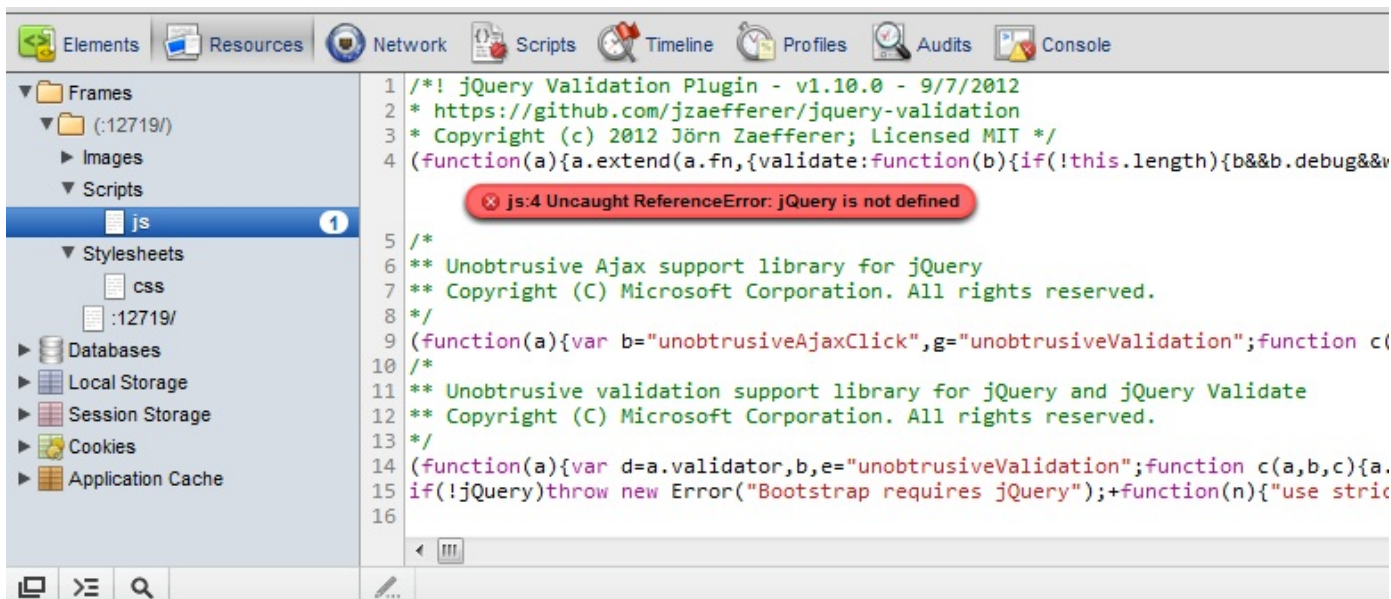
اگر واقعا مشکلی وجود داشته باشد، در همین لینک‌های پویایی که ارائه دادید، با کلیک بر روی آن‌ها عبارت «Minification failed. Returning unminified contents» ظاهر خواهد شد ( [یک نمونه مثال در اینجا](#) ) و سپس در سطرهای بعدی علت خطای کامپایل ذکر می‌شود که باید برطرف شود.

نویسنده: سیروان عفیفی  
تاریخ: ۰:۴۱ ۱۳۹۲/۰۵/۱۹

حاصل کلیک بر روی لینک‌های پویا تولید شده به صورت زیر است :  
مثلا با کلیک بر روی css تولید شده فایل Style.css را مشاهده میکنم :



و با کلیک بر روی فایل js تولید شده نیز به صورت زیر می‌باشد :



نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۵/۱۹ ۰۵:۵۸

چند سؤال اینجا هست که از راه دور قابل بررسی نیست:

- چطور تعاریف فایل‌ها رو در کدها اضافه کردید؟ (اگر خطای نبود jQuery رو بدون خطای کامپایل که ذکر شد گرفتید، یعنی اصلا jQuery تعریف نشده در تعریف bundle نهایی)
- ساختار پوشه‌های شما چیست؟ محل قرارگیری این فایل‌ها به چه ترتیبی است؟
- از چه نگارشی از اسکریپت‌ها و فایل‌ها استفاده می‌کنید؟ از کجا تهیه شده‌اند؟

ضمناً هیچ الزامی برای استفاده از minification در این مبحث نیست. فقط برای تکمیل بحث عنوان شده. به علاوه در قسمت‌های بعدی این بحث، سورس هر قسمت [پیوست شده](#) که شامل فایل‌های اسکریپت و CSS متناظر با هر بحث می‌شود. این فایل‌ها در این سری ویژه، مورد تأیید هستند و آزمایش شده‌اند.

نویسنده: فرزاد  
تاریخ: ۱۳۹۲/۰۶/۱۸ ۱۱:۱۲

سلام؛ بعد از اضافه کردن کلاس AsIsBundleOrderer به Bundleconfig با خطای:

```
'MvcApplication7.Helper.AsIsBundleOrderer' does not implement interface member
'System.Web.Optimization.IBundleOrderer.OrderFiles(System.Web.Optimization.BundleContext,
System.Collections.Generic.IEnumerable<System.Web.Optimization.BundleFile>')
```

مواجهه میشم. در واقع از کلاس AsIsBundleOrderer ارور میگیره.  
ممنون میشم راهنمایی کنید.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۶/۱۸ ۱۱:۲۷

- در نگارش جدید System.Web.Optimization، امضای FileInfo متد OrderFiles تبدیل شده به BundleFile. (همین تغییر نام کافی است)
- در زمان نگارش این مطلب، FileInfo بوده.



نویسنده: احسان شیروان  
تاریخ: ۱۷:۳۲ ۱۳۹۲/۰۷/۰۴

با سلام ضمن تشکر از شما بابت آموزش‌های بسیار خوب  
یه سوال داشتم من از C#.net , VS2010 دارم استفاده می‌کنم برای استفاده کردن از Bootstrap تمامی مراحل یاد شده توسط  
شما رو مو به مو اجرا کردم اما متأسفانه به هیچ وجه کار نداد البته خطا هم نمیده اما مباحث مربوط به ریسپانسیو بودن صفحه و  
حتی نمایش spanها جوابی نگرفتم آیا این بدلیل ورژن VS2010 هست یا دات نت؟ یا یه چیز دیگه ؟  
ظاهراً سایتی که ایجاد کردم هیچ درکی از span نداره در حالی که تمام فایل‌های bootstrap رو طبق آموزش روی سایت نصب  
کردم  
ممنون میشم راهنماییم نمایید

نویسنده: وحید نصیری  
تاریخ: ۱۷:۵۷ ۱۳۹۲/۰۷/۰۴

- مطالبی که من در مورد بوت استرپ منتشر کردم مربوط به نگارش 2 آن است. نگارش 3 آن (که مدتی است منتشر شده و از  
طریق لینک‌های ابتدای بحث به صورت پیش فرض دریافت می‌شود) دگرگون شده و خیلی از مباحث آن تغییر کرده‌اند. فایل‌های  
بوت استرپ 2 در قسمت‌های بعدی این بحث، [پیوست شده‌اند](#) که شامل فایل‌های اسکریپت و CSS متناظر با هر بحث می‌شود.  
+ [راهنمای مهاجرت از بوت استرپ 2 به 3](#)

نویسنده: صابر فتح الهی  
تاریخ: ۱۶:۴۴ ۱۳۹۲/۰۹/۱۰

سلام؛ من توی ASP.NET Webform از قابلیت Bundle استفاده کردم (قبلاً درست کار می‌کرد) اما توی نسخه 4.5 کار نمی‌کنه، بعد از  
دیباگ فهمیدم اصلاً فایلها با BundleCollection اضافه نمیشه و زمانی که می‌خواد خروجی بگیره با خطای نبود مدخل مورد نظر  
مواجه میشه. Update-Package را هم اجرا کردم اما همون خطا رو دریافت می‌کنم.

نویسنده: وحید نصیری  
تاریخ: ۱۸:۴۳ ۱۳۹۲/۰۹/۱۰

جزئیات روش خودتان را با روش یکی از اعضای تیم دات نت مطابقت بدید (مربوط به ASP.NET 4.5 است):  
[Adding Bundling and Minification to Web Forms](#)

نویسنده: افشار محبی  
تاریخ: ۱۷:۳۷ ۱۳۹۲/۰۹/۲۰

باید span را به col-md عوض کنید. همان طور که در راهنمای مهاجرت از ۲ به ۳ که لینکش را آقای نصیری مشخص کرده اند  
توضیح داده شده:

replace .span\* with .col-md-\*

بنده هم دقایقی را معطل کشف این تغییر بوت استرپ شدم.

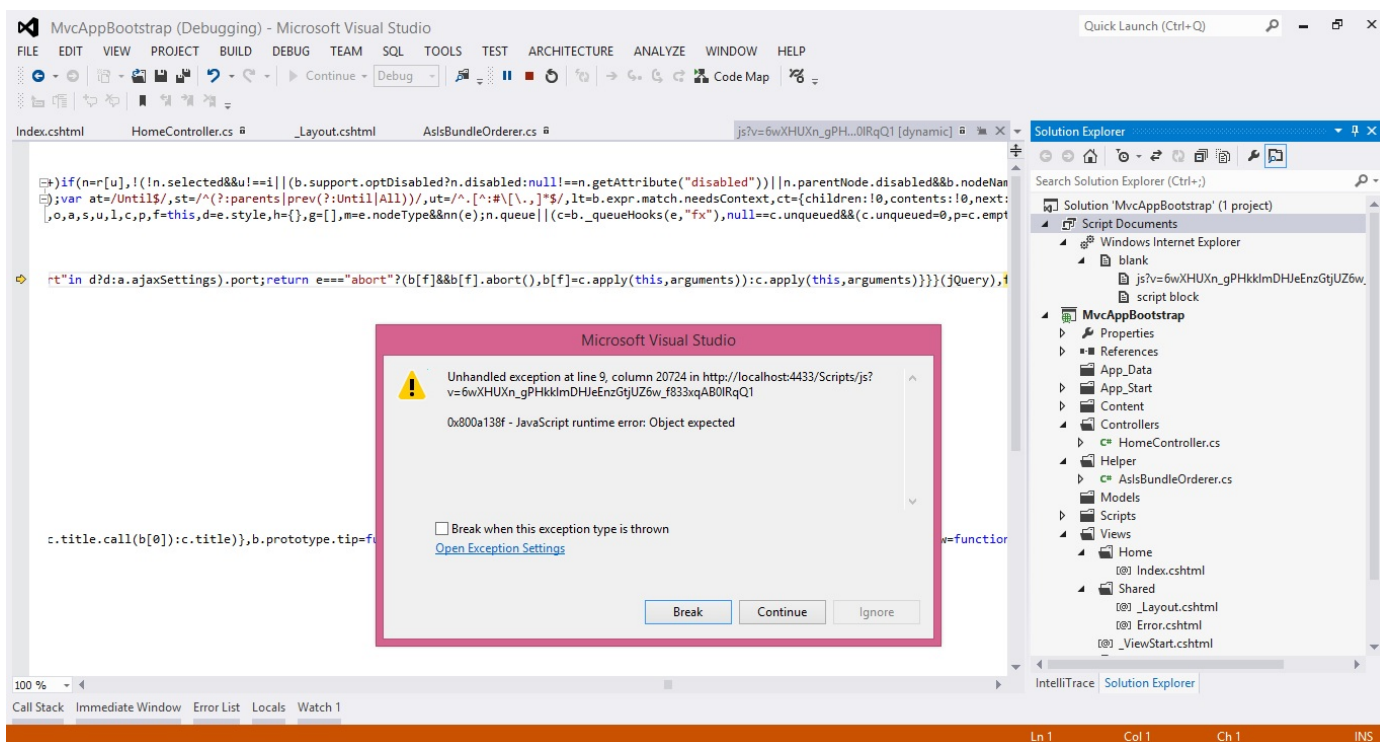
نویسنده: وحید نصیری  
تاریخ: ۱۷:۴۵ ۱۳۹۲/۰۹/۲۰

توضیحات تکمیلی در اینجا: « [بررسی سیستم جدید گرید بوت استرپ 3](#) »

نویسنده: مجتبی فخاری  
تاریخ: ۲۳:۰۵ ۱۳۹۲/۱۲/۰۳

بعد از نصب از طریق دستور Install-Package Twitter.BootstrapRTL و بعد هم دستور Install-Package PM>

Microsoft.AspNet.Web.Optimization بسته‌ها رو نصب کردم و یک پوشه به اسم Helper نیز اضافه نمودم و کلاسهای AsIsBundleOrderer را اضافه نمودم و در layout هم کدهای شما را آوردم ولی حالا در زمان اجرا خطای زیر را میدهد؟



میشه راهنمایی بفرمائید؟ با تشکر

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۱۲/۰۳ ۲۳:۱۳

مطلب جاری و سری مرتبط با آن برای بوت استرپ 2 نوشته شد. امروز اگر دستور `install-Package Twitter.BootstrapRTL` را صادر کنید، بوت استرپ 3.1 را نصب می‌کند که سیستم آن در بسیاری از موارد 180 درجه دگرگون شده نسبت به نگارش 2. بنابراین برای کار با نگارش 2 بهتر است دستور زیر را صادر کنید:

```
PM> Install-Package Twitter.BootstrapRTL -Version 2.3.2.1
```

و یا از مطالب [ارتقاء به بوت استرپ 3](#) کمک بگیرید.  
ضمناً فایل‌های مرتبط با نگارش 2، در قسمت‌های بعدی همین سری، در نظرات هر قسمت [پیوست شده‌اند](#).

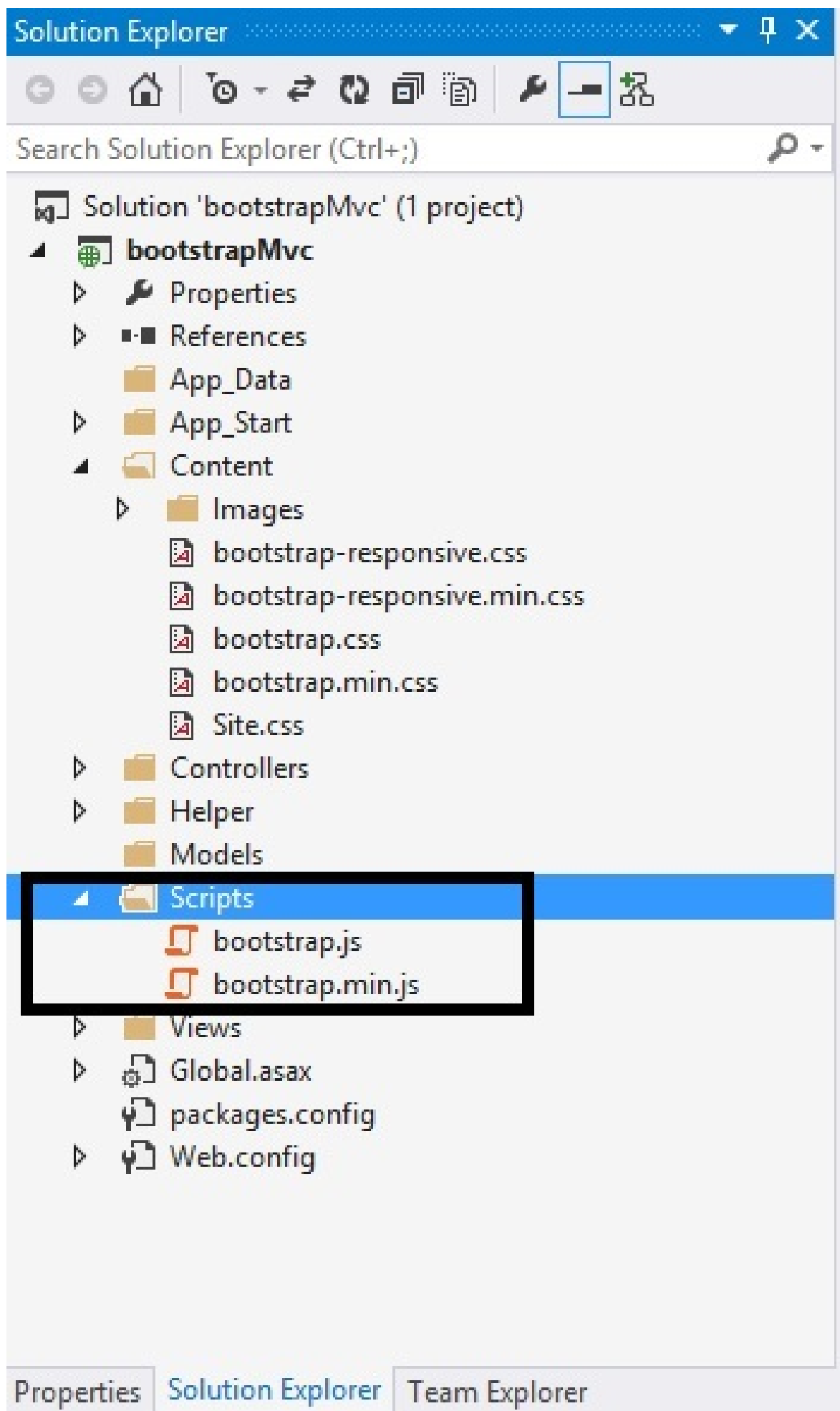
نویسنده: ایمان باقری  
تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۳:۲۵

سلام. من توی پروژم تو قسمت اسکریپت‌ها فقط `bootstrap.js` و `bootstrap.min.js` رو دارم و مابقی جاوا اسکریپت‌ها و جی کوئری هارو ندارم. با چه دستوری میتونم اونارو اضافه کنم. چون الان منوهای پایین افتادنی و یکسری چیزها دیگه کار نمیکنه. در ضمن پروژه خالی mvc شروع کردم. (تو حالت basic خودش پیش فرض داره)

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۳:۲۸

[کمی بالاتر](#) پاسخ دادم «مطلب جاری و سری مرتبط با آن برای بوت استرپ 2 نوشته شد ...». (چندین بار)

من هم بوت استرپ 2 رو گرفتم فقط فایل‌های جی کوئری و validation اون رو ندارم  
فایل‌های jquery-1.9.1.min.js jquery.unobtrusive-ajax.min.js jquery.validate.min.js و  
jquery.validate.unobtrusive.min.js



نویسنده:

وحید نصیری

تاریخ:

۱۳:۴۱ ۱۳۹۳/۰۱/۱۹

عنوان کردید حالت Basic پروژه‌های MVC 5 را شروع کردید. پیش فرض استاندارد MVC 5 [بوت استرپ 3](#) است و نه دو. فایل‌های کامل [بوت استرپ 2](#) را از طریق دستور ذیل که پیشتر نیز ذکر شده بود دریافت کنید (تمام اجزای js آن با هم یکی شده‌اند در قالب یک فایل نهایی):

```
PM> Install-Package Twitter.BootstrapRTL -Version 2.3.2.1
```

نویسنده:

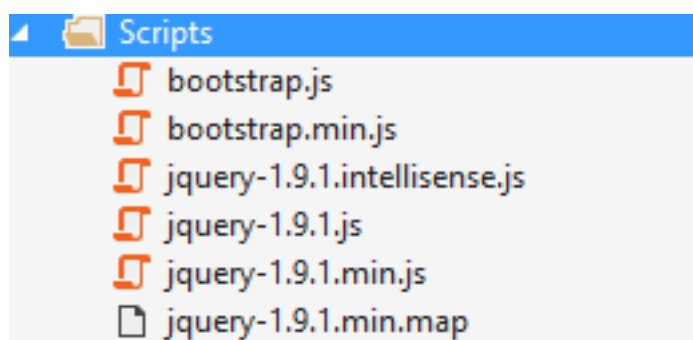
ایمان باقری

تاریخ:

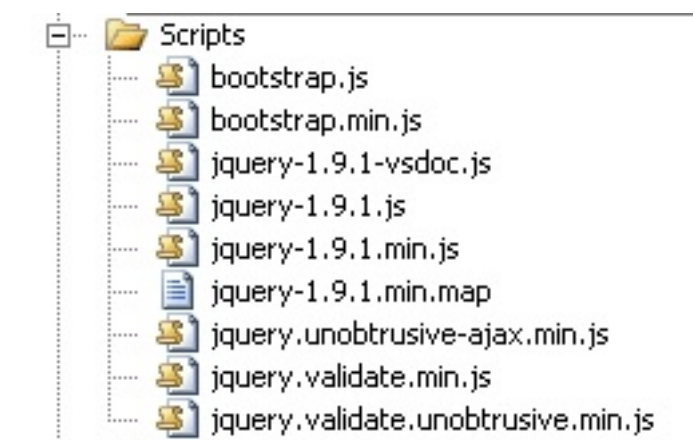
۱۴:۳ ۱۳۹۳/۰۱/۱۹

من حالت empty پروژه 4 mvc رو شروع کردم (بالا هم عرض کردم) و بوت استرپ هم از سایت rbootstrap.ir نسخه 2.3.2 v رو گرفتم.

البته الان که از دستور PM> Install-Package Twitter.BootstrapRTL -Version 2.3.2.1 استفاده کردم jquery-1.9.1.min.js رو هم نصب کرد



و مشکل منوها حل شد. فقط فایل‌های validation و ajax رو همچنان ندارم



نویسنده:

وحید نصیری

تاریخ:

۱۴:۱۲ ۱۳۹۳/۰۱/۱۹

فایل‌های دیگری را که یاد کردید، جزو بسته‌ی بوت استرپ نیستند. آن‌ها را دستی از یک پروژه‌ی MVC دیگر کپی کنید. از نیوگت هم قابل دریافت هستند: ( [^](#) و [^](#) )

نویسنده: M.Shakeri

تاریخ: ۱۴:۲۱ ۱۳۹۳/۰۶/۲۱

شما اشاره کردید به این که اگر پروژه رو در حالت release قرار بدیم فشرده سازی اعمال میشه یعنی اینکه در فایل web.config در قسمت compilation خصوصیت debug رو حذف کنیم:

```
<compilation debug="true" targetFramework="4.5" />
```

این کارو انجام دادم اما وقتی سورس صفحه رو نگاه می‌کنم فشرده سازی اعمال نشده به فایل‌های css و js !

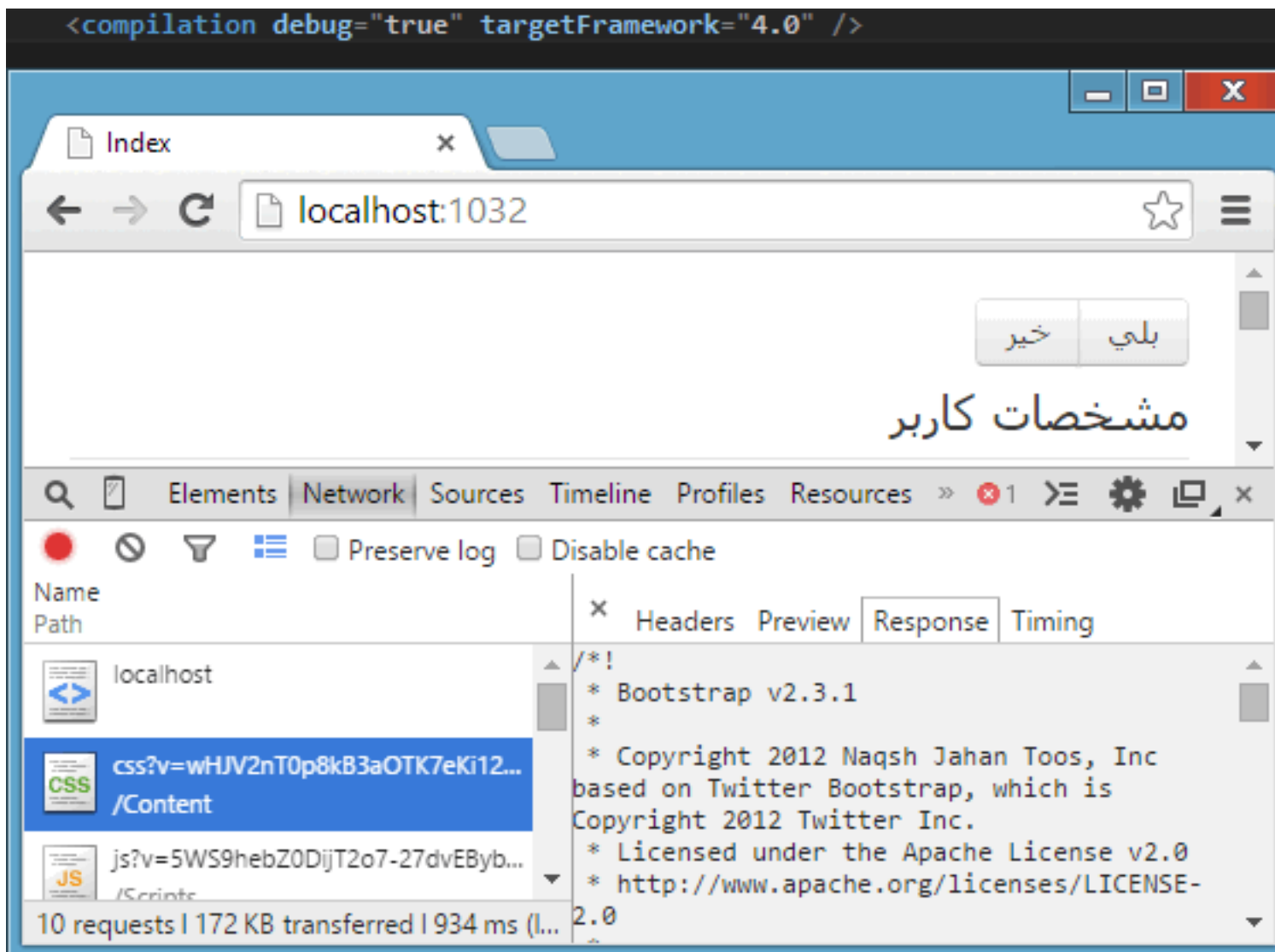
نویسنده: وحید نصیری

تاریخ: ۱۴:۴۲ ۱۳۹۳/۰۶/۲۱

پس از بررسی مجدد، مشکلی مشاهده نشد:  
- رندر حالت فشرده شده در حالت غیر دیباگ:



- رندر حالت معمولی و اصل آن در حالت دیباگ:



در کل اگر این روش برای شما کار نمی‌کند، شرط `HttpContext.Current.IsDebuggingEnabled` را از متد `addBundle` حذف کنید و `else` آنرا نگه دارید.

پس از آشنایی مقدماتی با Twitter Bootstrap، در این قسمت قصد داریم تا با ویژگی‌هایی از آن آشنا شویم که در کارهای روزمره طراحی وب بسیار مورد استفاده هستند؛ مانند تایپوگرافی، جداول، فرم‌ها، دکمه‌ها، تصاویر و آیکون‌ها.

## تایپوگرافی

هدف از تایپوگرافی، چیدمان متن به نحوی است که واضح، خوانا و مشخص باشد؛ همچنین مباحث زیبایی ارائه را نیز به آن اضافه کنید. برای مثال تنظیم فاصله بین حروف و کلمات، فاصله بین خطوط و یا رعایت یک سری نسبت‌های ویژه مانند [نسبت طلایی](#) جهت دعوت خواننده به مطالعه مطالب، بجای فراری دادن او، در مباحث تایپوگرافی رعایت می‌شوند. خوشبختانه Twitter Bootstrap به همراه یک سری تنظیمات تایپوگرافی پیش فرض است که در ادامه آن‌ها را مرور خواهیم کرد. پیش فرض‌های ابتدایی آن‌را مانند قلم با اندازه 14px، قلم پیش فرض Helvetica، فاصله بین خطوط و رنگ متن را در فایل bootstrap.css می‌توانید مشاهده کنید:

```
body {
margin: 0;"Helvetica Neue", Helvetica, Arial, sans-serif;
color: #333333;
background-color: #ffffff;
}
```

در اینجا اثر تنظیمات bootstrap را بر روی تگ‌های h1 تا h6 ملاحظه می‌کنید:

**سرتیتر 1**  
**سرتیتر 2**  
**سرتیتر 3**  
**سرتیتر 4**  
**سرتیتر 5**  
**سرتیتر 6**

```
<div class="row-fluid">
  <div class="span12">
    <h1>
      سرتیتر 1
    </h1>
    <h2>
      سرتیتر 2
    </h2>
    <h3>
      سرتیتر 3
    </h3>
    <h4>
      سرتیتر 4
    </h4>
    <h5>
      سرتیتر 5
    </h5>
    <h6>
      سرتیتر 6
    </h6>
  </div>
</div>
```

همچنین اگر نیاز باشد تا نسبت به متنی جلب توجه خواننده را جلب کرد می‌توان از کلاس lead استفاده نمود. به علاوه bootstrap پیش فرض‌هایی را به المان‌های small، strong و em نیز اعمال می‌کند:

```
<div class="row-fluid">
  <div class="span12">
    <p class="lead">
      تیتر
    </p>
```



```
<p>
متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن
متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن
متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن
متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن
</p>
<small>اندازه کوچک</small> <strong>متن ضخیم</strong> <em>متن ایتالیک</em>
</div>
</div>
```

## تیترا

متن  
متن  
متن  
متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن

اندازه کوچک متن ضخیم متن ایتالیک

روش دیگر جلب توجه به یک متن در bootstrap، استفاده از کلاس‌های text مانند text-error و امثال آن می‌باشد:

<pre>&lt;div class="row-fluid"&gt;   &lt;div class="span12"&gt;     &lt;p class="muted"&gt;       متن غیرفعال     &lt;p class="text-warning"&gt;       نمایش اخطار به کاربر     &lt;p class="text-error"&gt;       نمایش خطا به کاربر     &lt;p class="text-info"&gt;       نمایش اطلاعات به کاربر     &lt;p class="text-success"&gt;       نمایش موفقیت آمیز بودن     عملیات   &lt;/div&gt; &lt;/div&gt;</pre>	<p>متن بی صدا</p> <p>نمایش اخطار به کاربر</p> <p>نمایش خطا به کاربر</p> <p>نمایش اطلاعات به کاربر</p> <p>نمایش موفقیت آمیز بودن عملیات</p>
---	--

bootstrap تنظیماتی را به المان abbreviation موجود در HTML 5 نیز اعمال می‌کند. در این حالت کاربر با نزدیک ساختن اشاره‌گر ماوس به متن مشخص شده، یک tooltip توضیح دهنده نیز ظاهر خواهد شد:

<pre>&lt;div class="row-fluid"&gt;   &lt;div class="span12"&gt;     &lt;p&gt;       My       &lt;abbr title="منظور واحد پردازش مرکزی است"&gt;         CPU&lt;/abbr&gt;       has N Cores.     &lt;/p&gt;   &lt;/div&gt; &lt;/div&gt;</pre>	<p>My CPU has N Cores</p> <p>منظور واحد پردازش مرکزی است</p>
--	--

در bootstrap المان آدرس نیز شیوه نامه خاص خودش را داشته و به صورت یک قطعه خاص ظاهر می‌شود:

```
<div class="row-fluid">
  <div class="span12">
    <address>
      وحید نصیری
      <br />
      ایران، تهران
      <br />
      <abbr title="Phone">
        P: </abbr>
        12345678
      <br />
      <abbr title="Cell">
        C: </abbr>
        12345678
      </address>
    </div>
  </div>
```

وحید نصیری  
ایران، تهران  
P: 12345678  
C: 12345678  
Cell

در اینجا تاثیر bootstrap را بر روی المان blockquote ملاحظه می‌کنید؛ همچنین به همراه المان cite برای ذکر مآخذ نقل قول ذکر شده:

```
<div class="row-fluid">
  <div class="span12">
    <blockquote>
      <p>
        جهت نمایش نقل قول
      </p>
      <small><cite title="کاربر">
        شماره 2</cite> از شخصی
      </small>
    </blockquote>
  </div>
</div>
```

جهت نمایش نقل قول  
— از شخصی  
کاربر شماره ۲

در bootstrap برای حذف بولت‌های کنار یک لیست مرتب، فقط کافی است از کلاس unstyled استفاده شود:

```
<div class="row-fluid">
  <div class="span6">
    <ul class="unstyled">
      <li>یک </li>
      <li>دو </li>
      <li>سه </li>
    </ul>
  </div>
  <div class="span6">
    <ol>
      <li>یک </li>
      <li>دو </li>
      <li>سه </li>
    </ol>
  </div>
</div>
```

1. يك  
2. دو  
3. سه

به علاوه امکان تعریف دو نوع خاص از definition list (المان dl در اینجا). در حالت عادی، ابتدا عنوان مشخص شده با dt یا definition term به صورت ضخیم ظاهر می‌شود؛ به همراه توضیحات ارائه شده در المان dd آن در سطر بعدی. اگر از کلاس dl-horizontal استفاده شود، همانند تصویر ذیل، عنوان در کنار توضیحات در یک سطر قرار خواهد گرفت:

```
<div class="row-fluid">
  <div class="span12">
    <dl class="dl-horizontal">
      <dt>عنوان</dt>
```

## عنوان توضیحات بلند در اینجا

```
<dd>
    توضیحات بلند در اینجا
</dd>
</div>
</div>
```

در bootstrap امکان اعمال شیوه نامه ابتدایی به کدهای برنامه‌ها نیز وجود دارد. اگر از تگ code استفاده شود، فرض بر این است که قرار است اطلاعاتی را در یک سطر نمایش دهید. اگر اطلاعات کدها، بیشتر از یک سطر است، می‌توان از تگ pre استفاده کرد. در اینجا با اعمال کلاس pre-scrollable به تگ pre، به صورت خودکار یک اسکرول عمودی به قطعه کدها اعمال خواهد شد:

```
<div class="row-fluid">
  <div class="span6">
    <code>inline code</code>
  </div>
  <div class="span6">
    <pre class="pre-scrollable">
      code
      code
      code
    </pre>
  </div>
</div>
```



## استفاده از جداول و تاثیر bootstrap بر آنها

در ادامه کدهای یک جدول متداول را که مزین شده‌است به کلاس‌های bootstrap ملاحظه می‌کنید:

```
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span12">
      <table
class="table table-striped table-hover table-bordered table-condensed">
        <caption>
          عنوانی خاص در اینجا
        </caption>
        <thead>
          <tr>
            <th>
              ستون یک
            </th>
            <th>
              ستون دو
            </th>
            <th>
              ستون سه
            </th>
          </tr>
        </thead>
        <tbody>
          <tr class="error">
            <td>
              متن یک 1
            </td>
            <td>
              متن دو 1
            </td>
            <td>
              متن سه 1
            </td>
          </tr>
          <tr>
            <td>
              متن یک 2
            </td>
            <td>
              متن دو 2
            </td>
            <td>
              متن سه 2
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
```

```

        </td>
        <td>
            متن سه 2
        </td>
    </tr>
    <tr>
        <td>
            متن یک 3
        </td>
        <td>
            متن دو 3
        </td>
        <td>
            متن سه 3
        </td>
    </tr>
</tbody>
</table>
</div>
</div>
</div>

```

عنوانی خاص در اینجا

ستون يك	ستون دو	ستون سه
1 متن يك	1 متن دو	1 متن سه
2 متن يك	2 متن دو	2 متن سه
3 متن يك	3 متن دو	3 متن سه

در اینجا ذکر caption اختیاری است. وجود thead و tbody به تشخیص هدر و ردیف‌ها جهت اعمال شیوه‌نامه‌های متناظر کمک می‌کنند. همچنین کلاس‌های ذیل نیز به جدول اعمال شده‌اند:

table: سبب می‌شود تا تنظیمات ابتدایی bootstrap به جدول طراحی شده، اعمال شوند.

table-striped: رنگ زمینه سطرها را یک در میان تغییر می‌دهد.

table-hover: سبب می‌شود تا با عبور اشاره‌گر ماوس از روی سطرها، رنگ زمینه آن‌ها تغییر کنند.

table-bordered: حاشیه‌ای را به جدول و ردیف‌ها اعمال می‌کنند. همچنین سبب نمایش گوشه‌های گرد نیز می‌شود.

table-condensed: اندکی padding اعمال شده به سلول‌های جداول را کاهش می‌دهد و جدول را فشرده‌تر می‌کند.

در جدول فوق، کلاس نمونه error به یک tr نیز اعمال شده‌است تا اثر آن را بر روی یک ردیف بهتر بتوان ملاحظه کرد.

## طراحی فرم‌ها و تاثیر bootstrap بر آن‌ها

قرار دادن برجسب‌ها و عناصر صفحه، به نحوی کاربرپسند و دلپذیر، نیاز به رعایت یک سری اصول تایپوگرافی و طراحی ویژه دارد که این موارد نیز در bootstrap گنجانده شده‌اند.

### (1) فرم‌های عمودی

```

<div class="container-fluid">
    <div class="row-fluid">
        <div class="span12">
            <form action="/signup"
method="post">
                <fieldset>
                    <legend>ثبت نام</legend>
                    <label>
                        ایمیل:</label>
                    <input name="email"
type="text" />
                </fieldset>
            </div>
        </div>
    </div>
</div>

```



<pre> &lt;div&gt;     &lt;label class="radio"     inline"&gt;         &lt;input type="radio"         name="isMale" /&gt;         مذکر&lt;/label&gt;     &lt;label class="radio"     inline"&gt;         &lt;input type="radio"         name="isFemale" /&gt;         مؤنث&lt;/label&gt; &lt;/div&gt; </pre>	
---	--

#### (4) تعیین اندازه فیلدها

با استفاده از کلاس‌هایی مانند `input-mini`, `input-small`, `input-medium`, `input-large`, `input-xlarge` و `input-xxlarge` می‌توان اندازه فیلدها را کوچک‌تر از اندازه معمول یا بزرگتر کرد. یا حتی می‌توان از همان کلاس‌های `span` مانند `span12` (اختصاص کل عرض به یک فیلد) و امثال آن برای تعیین اندازه یک فیلد استفاده کرد. اگر علاقمند هستید که یک `textarea` کل عرض صفحه را به خود اختصاص دهد، از کلاس `input-block-level` استفاده کنید.

#### (5) فرم‌های جستجو

<pre> &lt;form class="search-form"&gt;     &lt;fieldset&gt;         &lt;legend&gt;جستجو&lt;/legend&gt;         &lt;input type="search"         class="search-query" /&gt;         &lt;button type="submit"         class="btn" &gt;بیاب&lt;/button&gt;     &lt;/fieldset&gt; &lt;/form&gt; </pre>	
---	--

چند نکته در فرم‌های جستجوی طراحی شده با bootstrap نسبت به بقیه فرم‌ها حائز اهمیت است:

- کلاس فرم بهتر است `search-form` تعیین شود
- نوع `input` بهتر است `search` وارد شود
- کلاس `input` جستجو نیز `search-query` انتخاب گردد

همانطور که ملاحظه می‌کنید، در این حالت گوشه‌های جعبه متنی جستجو، نسبت به حالت‌های معمولی آن‌ها گرد شده است. کلاس دکمه نیز `btn` در نظر گرفته شده است تا حالت ویژه دکمه‌های bootstrap را پیدا کند.

#### (6) فرم‌های افقی

تا اینجا، با فرم‌های حالت پیش فرض یا فرم‌هایی که عناصر را به صورت پشت‌پشتی عمودی بر روی یکدیگر قرار می‌دهند، آشنا شدیم. حالت متداول دیگر طراحی فرم‌ها، حالت افقی است. به این معنا که در هر سطر، یک برچسب و یک المان قرار گیرند، بجای اینکه ابتدا برچسب نمایش داده شود و در سطر بعدی، المان مرتبط با آن. Bootstrap برای طراحی بدون استفاده از جداول این نوع فرم‌ها نیز تنظیمات خاصی را تدارک دیده است.

<pre> &lt;form class="form-horizontal" action="/signup" method="post"&gt;     &lt;fieldset&gt; </pre>	
---	--

<pre> &lt;legend&gt;ثبت نام&lt;/legend&gt; &lt;div class="control-group"&gt;   &lt;label class="control-label"&gt;ایمیل:&lt;/label&gt;   &lt;div class="controls"&gt;     &lt;input name="email" type="text" /&gt;   &lt;/div&gt; &lt;/div&gt; &lt;div class="control-group"&gt;   &lt;label class="control-label"&gt;نام:&lt;/label&gt;   &lt;div class="controls"&gt;     &lt;input name="name" type="text" /&gt;   &lt;/div&gt; &lt;div class="control-group"&gt;   &lt;div class="controls"&gt;     &lt;input type="radio" name="isMale" /&gt;     &lt;label class="radio"&gt;مذکر&lt;/label&gt;     &lt;input type="radio" name="isFemale" /&gt;     &lt;label class="radio"&gt;مؤنث&lt;/label&gt;   &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; &lt;/form&gt; </pre>	
---	--

مطابق مثال فوق، خلاصه طراحی فرم‌های افقی با Bootstrap به این نحو است:

- کلاس form-horizontal را به فرم جاری اضافه کنید.
- هر سطر مورد نظر را در div ایی با کلاس control-group محصور نمایید.
- به برچسب‌ها، کلاس control-label را انتساب دهید.
- کنترل‌های مدنظر را در div ایی با کلاس controls محصور کنید.

هر چند این روش نیاز به اندکی HTML نویسی دارد، اما بسیاری به این نوع فرم‌ها بیشتر علاقمند هستند تا فرم‌های عمودی ابتدای بحث.

## 7) جلب توجه کاربران به فیلدها برای نمایش خطاهای اعتبارسنجی

<pre> &lt;div class="control-group error"&gt;   &lt;label class="control-label"&gt;ایمیل:&lt;/label&gt;   &lt;div class="controls"&gt;     &lt;input name="email" type="text" /&gt;     &lt;span class="help-block"&gt;لطفا ایمیل را با&lt;/span&gt;     &lt;span class="help-block"&gt;فرمت صحیحی وارد نمایید&lt;/span&gt;   &lt;/div&gt; &lt;/div&gt; </pre>	
--	--

	<div><div></div><div>را با فرمت صحیحی وارد نمائید</div><div></div><div>مؤت</div></div>
--	--

برای تزریق خطاهای اعتبارسنجی ویژه، در اینجا می‌توان از کلاس‌های `help-block` و `help-inline` به همراه کلاس‌هایی مانند `success` و `error`، `info`، `warning` استفاده کرد.

## 8) توسعه فیلدهای استاندارد

<pre>&lt;div class="input-prepend input-append"&gt;     &lt;span dir="ltr" class="add-on"&gt;.00&lt;/span&gt;     &lt;input dir="ltr" type="text" /&gt;     &lt;span class="add-on"&gt;ریال&lt;/span&gt; &lt;/div&gt;</pre>	<div><div>ریال</div><div>123</div><div>.00</div></div>
---	--

توسط ترکیب کلاس‌های `input-prepend`، `input-append` و `span` با کلاس `add-on`، می‌توان عناصری بصری خاصی را به عنوان جزئی از فیلد مورد نظر اضافه کرد.

## 9) HTML Helpers مخصوص ASP.NET MVC برای کار با bootstrap

نکاتی را که در اینجا مطرح شدند، اگر علاقمند بودید که به شکلی `strongly typed` در ASP.NET MVC اعمال کنید، می‌توان به پروژه‌هایی مانند `TwitterBootstrapMvc` مراجعه کرد. تعداد این نوع پروژه‌ها هم روز به روز بیشتر می‌شوند:

<https://twitterbootstrapmvc.codeplex.com>

<https://mvc4bootstraphelper.codeplex.com>

<https://github.com/erichexter/twitter.bootstrap.mvc>

<http://bootstraphelpers.codeplex.com>

تنظیمات خاص دکمه‌ها در حین استفاده از Twitter Bootstrap



در مثال ذیل، کلاس‌های مرتبط با تزئین دکمه‌ها را توسط bootstrap، ملاحظه می‌کنید:

```
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span12">
      <table
class="table table-striped table-hover table-bordered table-condensed">
        <thead>
          <tr>
            <th>
              دکمه
            </th>
            <th>
              لینک
            </th>
            <th>
              کلاس بکار گرفته شده
            </th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>
              <button class="btn">
                default</button>
            </td>
            <td>
              <a href="#" class="btn">default</a>
            </td>
            <td>
              <code>btn</code>
            </td>
          </tr>
          <tr>
            <td>
              <button class="btn btn-primary">
                primary</button>
            </td>
            <td>
              <a href="#" class="btn btn-primary">primary</a>
            </td>
            <td>
              <code>btn btn-primary</code>
            </td>
          </tr>
          <tr>
            <td>
              <button class="btn btn-info">
                info</button>
            </td>
            <td>
              <a href="#" class="btn btn-info">info</a>
            </td>
            <td>
              <code>btn btn-info</code>
            </td>
          </tr>
          <tr>
            <td>
              <button class="btn btn-success">
                success</button>
            </td>
            <td>
              <a href="#" class="btn btn-success">success</a>
            </td>
            <td>
              <code>btn btn-success</code>
            </td>
          </tr>
          <tr>
            <td>
              <button class="btn btn-warning">
                warning</button>
            </td>
            <td>
              <a href="#" class="btn btn-warning">warning</a>
            </td>
            <td>
              <code>btn btn-warning</code>
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
```

```

        </td>
      </tr>
      <tr>
        <td>
          <button class="btn btn-danger">
            danger</button>
        </td>
        <td>
          <a href="#" class="btn btn-danger">danger</a>
        </td>
        <td>
          <code>btn btn-danger</code>
        </td>
      </tr>
      <tr>
        <td>
          <button class="btn btn-inverse">
            inverse</button>
        </td>
        <td>
          <a href="#" class="btn btn-inverse">inverse</a>
        </td>
        <td>
          <code>btn btn-inverse</code>
        </td>
      </tr>
      <tr>
        <td>
          <button class="btn btn-link">
            link</button>
        </td>
        <td>
          <a href="#" class="btn btn-link">link</a>
        </td>
        <td>
          <code>btn btn-link</code>
        </td>
      </tr>
      <tr>
        <td>
          <button class="btn btn-primary btn-large">
            large</button>
        </td>
        <td>
          <a href="#" class="btn btn-primary btn-large">large</a>
        </td>
        <td>
          <code>btn btn-primary btn-large</code>
        </td>
      </tr>
      <tr>
        <td>
          <button class="btn btn-primary btn-small">
            small</button>
        </td>
        <td>
          <a href="#" class="btn btn-primary btn-small">small</a>
        </td>
        <td>
          <code>btn btn-primary btn-small</code>
        </td>
      </tr>
      <tr>
        <td>
          <button class="btn btn-primary btn-mini">
            mini</button>
        </td>
        <td>
          <a href="#" class="btn btn-primary btn-mini">mini</a>
        </td>
        <td>
          <code>btn btn-primary btn-mini</code>
        </td>
      </tr>
      <tr>
        <td>
          <button class="btn btn-primary btn-block">
            block</button>
        </td>
        <td>
          <a href="#" class="btn btn-primary btn-block">block</a>

```

```

</td>
<td>
<code>btn btn-primary btn-block</code>
</td>
</tr>
<tr>
<td>
<code>button class="btn btn-primary disabled">
disabled</button>
</td>
<td>
<a href="#" class="btn btn-primary disabled">disabled</a>
</td>
<td>
<code>btn btn-primary disabled</code>
</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>

```

دکمه	لینک	کلاس بکار گرفته شده
		<code>btn</code>
		<code>btn btn-primary</code>
		<code>btn btn-info</code>
		<code>btn btn-success</code>
		<code>btn btn-warning</code>
		<code>btn btn-danger</code>
		<code>btn btn-inverse</code>
		<code>btn btn-link</code>
		<code>btn btn-primary btn-large</code>
		<code>btn btn-primary btn-small</code>
		<code>btn btn-primary btn-mini</code>
		<code>btn btn-primary btn-block</code>
		<code>btn btn-primary disabled</code>

همانطور که ملاحظه کردید، الزامی ندارد که این کلاس‌ها را حتما به دکمه‌ها اعمال کرد. برای نمونه می‌توان از یک span یا لینک نیز برای تعریف دکمه‌ها بهره جست. برای یکپارچه سازی چنین دکمه‌هایی (که در اصل دکمه نیستند) با ASP.NET MVC می‌توان به مطلب تکمیلی « [استفاده از دکمه‌های CSS توئیت در ASP.NET MVC](#) » مراجعه نمود.

**یک نکته:** اگر علاقمند هستید که تعدادی دکمه را به شکل یک toolbar نمایش دهید، آن‌ها را در یک div محصور کرده و کلاس btn-group را به آن div اعمال نمائید.

## کار با تصاویر و آیکون‌ها در Twitter Bootstrap

کلاس‌هایی مانند img-rounded, img-circle, img-polaroid با اعمال به یک تصویر، سبب گردن شدن گوشه‌های آن، نمایش دایره‌ای و یا نمایش به همراه حاشیه یک تصویر خواهند شد. Twitter Bootstrap به همراه صدها آیکون ارائه شده است. این آیکون‌ها توسط [glyphicons.com](http://glyphicons.com) ایجاد شده‌اند. روشی که برای استفاده از آن‌ها توصیه شده است، استفاده از تگ i می‌باشد. برای مثال:

```
<i class="icon-music"></i>icon-music
```

البته بدیهی است که محدودیتی در اینجا وجود نداشته و می‌توان این کلاس‌ها را به یک span نیز اعمال کرد. برای مشاهده لیست کلاس‌های قابل استفاده، کلمه icon-glass را در فایل bootstrap.css جستجو نمائید؛ تا شروع مدخل مرتبط با آیکون‌ها را بتوانید مشاهده نمائید. رنگ پیش فرض این آیکون‌ها مشکی است. اگر علاقمند بودید که آن‌ها را برای مثال با رنگ سفید نمایش دهید فقط کافی است کلاس icon-white را پس از کلاس آیکون مدنظر، ذکر کرد:

```
<i class="icon-music icon-white"></i>icon-music
```

از این نمونه قلم‌های سازگار با Twitter Bootstrap در آدرس [fontawesome.github.com](http://fontawesome.github.com) نیز قابل دریافت هستند. امکان اعمال این آیکون‌ها به دکمه‌ها نیز وجود دارد. برای مثال:

```
<button class="btn">
  <i class="icon-music"></i>دکمه</button>
```

کاربرد دیگر این آیکون‌ها در بحث توسعه فیلدهای استاندارد است که پیشتر بحث شد. برای مثال نمایش آیکون نامه در کنار فیلد دریافت ایمیل:

<pre>&lt;div class="input-prepend"&gt;   &lt;span class="add-on"&gt;&lt;i class="icon-envelope"&gt;&lt;/i&gt;&lt;/span&gt;   &lt;input type="email" dir="ltr" /&gt; &lt;/div&gt;</pre>	
--	--

## نظرات خوانندگان

نویسنده: pezhman  
تاریخ: ۱۵:۲ ۱۳۹۲/۰۳/۱۸

سلام  
ممنون بابت آموزش  
می خواستم بدونم چطور از راست به چپ رو درست کردین؟ مخصوصا فرم ها رو  
آیا از pull-right استفاده کردین یا css دیگه؟  
(ممنون :)

نویسنده: وحید نصیری  
تاریخ: ۱۷:۵۱ ۱۳۹۲/۰۳/۱۸

من هیچ تغییر خاصی ندادم. فقط از بسته RTL [معرفی شده در قسمت طرحبندی](#) استفاده کردم.

نویسنده: \_mehdi  
تاریخ: ۶:۴۹ ۱۳۹۲/۰۳/۲۰

سلام  
تشکر بابت آموزشهاون  
شما برای تست صفحات وب در دستگاههای مختلف از جمله تبلت و موبایل اونهم با رزولیشنهای مختلف از چه ابزار یا نرم افزار استفاده میکنید.

نویسنده: وحید نصیری  
تاریخ: ۸:۴۵ ۱۳۹۲/۰۳/۲۰

- جواب شخصی: چنین کاری رو نمیکنم. چون مطابق آمار حداقل سایت جاری حدود یک درصد یا کمتر از بازدیدکنندگان این سایت کاربر غیر PC هستند که عملا در محاسبات به حساب نمیان. (واقع گرایی)  
- جواب کلی: یک سری شبیه ساز رو [از اینجا](#) میتونید دریافت کنید. [امکان یکپارچه سازی](#) با VS.NET هم وجود دارد.

پس از آشنایی با [طرح‌بندی صفحات](#) و [امکانات متداول](#) Twitter Bootstrap، در این قسمت به کامپوننت‌ها و اجزای تعاملی آن مانند منوها، برگه‌ها و امثال آن خواهیم پرداخت.

### منوهای پایین افتادنی (Dropdown menus) در Twitter Bootstrap

برای کار با منوها، حداقل نیازهایی که باید به صفحه اضافه شوند، فایل css مرتبط با Twitter Bootstrap، فایل اسکریپت‌های جی‌کوئری و فایل bootstrap.min.js است. در ادامه، ساختار متداول یک منوی نمونه ایجاد شده را ملاحظه می‌کنید:

```
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span12">
      <div class="dropdown">
        <a class="dropdown-toggle" data-toggle="dropdown" href="#">منوی پایین افتادنی</a>
        <span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="http://www.site1.com">site1</a></li>
          <li><a href="http://www.site2.com">site2</a></li>
          <li><a href="http://www.site3.com">site3</a></li>
          <li class="divider"></li>
          <li class="dropdown-submenu"><a href="#">سایر موارد</a>
            <ul class="dropdown-menu">
              <li><a href="http://www.site4.com">site4</a></li>
              <li><a href="http://www.site5.com">site5</a></li>
            </ul>
          </li>
        </ul>
      </div>
    </div>
  </div>
</div>
```



در اینجا به یک div، کلاس dropdown نسبت داده شده است؛ از این جهت که می‌توان این منو را به انواع و اقسام المان‌های صفحه

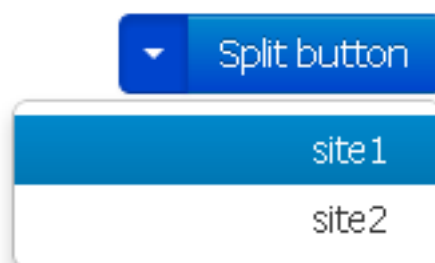
نیز نسبت داد. به این ترتیب، منو از محل قرارگیری این المان باز خواهد شد. در ادامه روال متداول تعریف منو را در Bootstrap، ملاحظه می‌کنید. کلاس dropdown-toggle سبب خواهد شد تا با کلیک ماوس بر روی لینک، منو ظاهر شود. کلاس caret نیز سبب نمایش مثلث رو به پایین، جهت مشخص سازی وجود منو در این محل خاص، تعریف شده است. المان‌های منو، توسط لیست‌ها تعریف می‌شوند. این لیست باید با کلاس dropdown-menu شروع شود. سپس هر عضو این لیست، یک آیتم منو را تشکیل خواهد داد. اگر نیاز به خط جدا کننده‌ای در این میان باشد، باید از کلاس divider استفاده کرد. در اینجا همچنین امکان تعریف منوهای تو در تو نیز وجود دارد. برای اینکار تنها کافی است یک لیست دیگر را در این میان اضافه کرد. کلاس dropdown-submenu به عضو لیست دربرگیرنده زیر منو، اضافه می‌شود. زیر منو نیز دارای کلاس dropdown-menu خواهد بود.

### چند نکته تکمیلی در مورد منوها:

- همانطور که در قسمت بررسی دکمه‌های Bootstrap نیز عنوان شد، برای تبدیل یک المان به دکمه فقط کافی است کلاس btn را به آن انتساب دهیم. در اینجا نیز می‌توان کلاس لینک منوی پایین افتادنی را به صورت btn btn-primary dropdown-toggle کرد تا ظاهر آن تبدیل به یک دکمه Bootstrap شود.
- در اولین div مثال فوق، کلاس dropdown ذکر شده است. می‌توان این کلاس را به dropup تغییر داد تا نحوه باز شدن منوی تعریف شده رو به بالا باشد؛ بجای رو به پایین.
- برای ساخت split button که تشکیل شده است از دو دکمه اصلی و دکمه کوچکی در کنار آن که یک زیرمنو را نمایش می‌دهد، باید به نحو زیر عمل کرد:

```
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span12">
      <div class="btn-group">
        <a class="btn btn-primary" href="#">Split button</a> <a class="btn btn-primary
dropdown-toggle"
        data-toggle="dropdown" href="#"><span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="http://www.site1.com">site1</a></li>
          <li><a href="http://www.site2.com">site2</a></li>
        </ul>
      </div>
    </div>
  </div>
</div>
```

تنها نکته‌ای که در اینجا نسبت به حالت قبل اضافه شده است، استفاده از کلاس btn-group است. این کلاس سبب می‌شود تا دکمه‌های تعریف شده در کنار هم به شکل یک toolbar مرتب شوند.



### تعریف برگه‌ها در Twitter Bootstrap

برای تعریف منوهای راهنمای سایت به شکل برگه، کار با استفاده از کلاس nav و تعریف یک لیست ساده شروع می‌شود:

```

<div class="container">
  <div class="row">
    <h2>
      Navigational menus
    </h2>
    <div class="span3">
      <ul class="nav nav-tabs">
        <li class="nav-header">nav nav-tabs</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
      </ul>
    </div>
    <div class="span3">
      <ul class="nav nav-tabs nav-stacked">
        <li class="nav-header">nav nav-tabs nav-stacked</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
      </ul>
    </div>
    <div class="span3">
      <ul class="nav nav-pills">
        <li class="nav-header">nav nav-pills</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
      </ul>
    </div>
    <div class="span3">
      <ul class="nav nav-list">
        <li class="nav-header">nav nav-list</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
        <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown"
href="#">Menu
          ...<span class="caret"></span></a>
          <ul class="dropdown-menu">
            <li><a href="http://www.site1.com">site1</a></li>
            <li><a href="http://www.site2.com">site2</a></li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</div>

```



# Navigational menus

Login

Contact

Home

NAV NAV-TABS

NAV NAV-TABS NAV-STACKED

Home

Contact

Login

Login

Contact

Home

NAV NAV-PILLS

NAV NAV-LIST

Home

Contact

Login

... Menu

site1

site2

- اضافه شدن کلاس nav، بولت‌های لیست را حذف و ویژگی تغییر رنگ زمینه هر یک از عناصر لیست را با حرکت اشاره‌گر ماوس بر روی آن فعال می‌کند.
- برای اعمال اشکال مختلف به کلاس nav، می‌توان از سایر کلاس‌های مرتبط استفاده کرد. اگر از nav-tabs استفاده شود، این لیست عمودی تبدیل به یک لیست افقی خواهد شد. اگر کلاس active به یکی از آیتم‌های آن نسبت داده شود (بیانگر صفحه جاری)، این گزینه حالت انتخاب شده را پیدا می‌کند.
- در هر حالت، برای تبدیل چیدمان افقی به عمودی، از کلاس nav-stacked استفاده نمایید.
- کلاس nav-pills، برای نمایش عنصر فعال، از یک مستطیل با گوشه‌های گرد استفاده می‌کند.
- اگر می‌خواهید این گوشه‌ها گرد نباشند، از nav-list استفاده کنید.
- در هر کدام از حالت‌ها می‌توان از یک گزینه اختیاری با کلاس nav-header برای نمایش متنی خاکستری بالای منوی راهبری استفاده کرد.
- همانطور که در آخرین لیست اضافه شده در مثال فوق ملاحظه می‌کنید، امکان تعریف منوهای پایین افتادنی در اینجا نیز میسر است. در این حالت کلاس li تعریف شده باید dropdown تعریف شود. مابقی نکات آن مانند قبل است.

## تعریف یک Tab به همراه محتوای برگه‌های آن در داخل یک صفحه

در حالت تعریف منوهای راهبری فوق، هر عنصر برگه‌های تعریف شده به یک صفحه جدید لینک شده بود. اگر نیاز باشد تا هر برگه، محتوایی داخل همان صفحه داشته باشد باید به نحو ذیل عمل کرد:

```
<div class="container">
  <div class="row">
    <div class="span12">
      <div class="tabbable">
        <ul class="nav nav-tabs">
          <li class="active"><a data-toggle="tab" href="#tab1">Home</a></li>
          <li><a data-toggle="tab" href="#tab2">About</a></li>
        </ul>
        <div class="tab-content">
          <div class="tab-pane active" id="tab1">
            data ... data ...
          </div>
          <div class="tab-pane" id="tab2">
            data2 ... data2 ...
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

About

Home

... data2 ... data2

در اینجا کار با کلاس tabbable شروع می‌شود. سپس از یک لیست با کلاس nav nav-tabs برای تعریف سرتیترهای برگه‌ها استفاده می‌کنیم.

اگر به مثال فوق دقت کنید، اینبار hrefهای تعریف شده به tab1 و tab2 اشاره می‌کنند. ایندو در حقیقت id محتوای متناظر هستند که در قسمت div ایی با کلاس tab-content به div هایی با id مساوی tab1 و tab2 انتساب داده شده‌اند.

## شیوه دوم تعریف منوی راهبری سایت به کمک Twitter Bootstrap

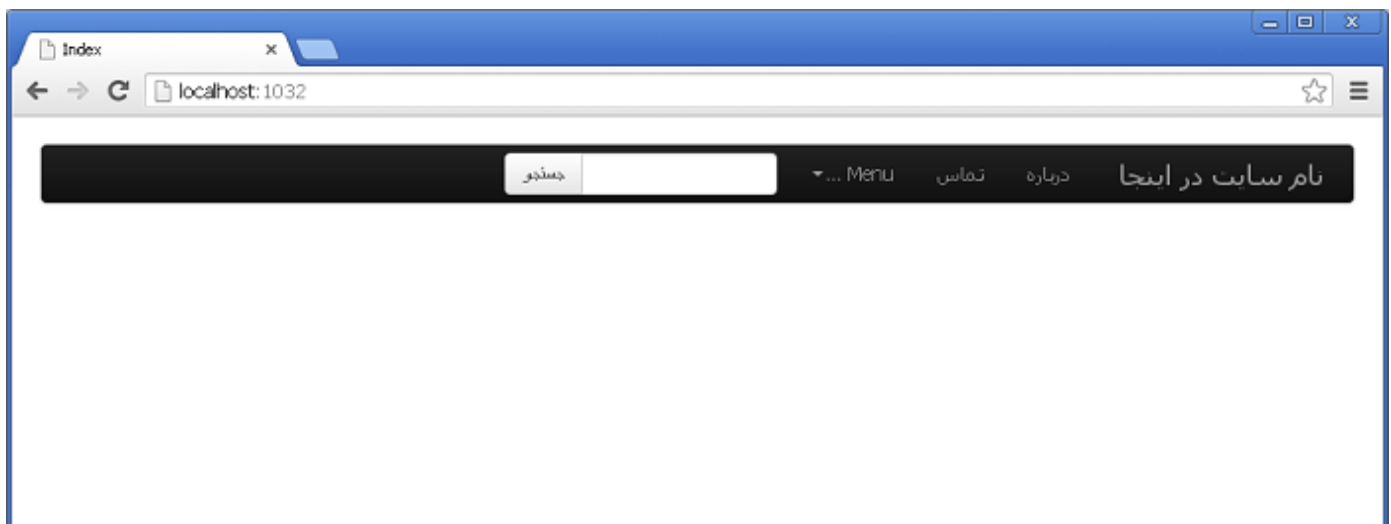
اگر علاقمند باشید که منوی راهبری بالای سایت دارای یک حاشیه به همراه مسایل طراحی واکنش‌گرا باشد، می‌توان از کامپوننت navbar مجموعه Bootstrap استفاده کرد:

```
<div class="container">
  <div class="row">
    <div class="span12">
      <nav class="navbar navbar-inverse">
        <div class="navbar-inner">
          <a href="/" class="brand">نام سایت در اینجا</a>
          <a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </a>
        </div>
      </nav>
    </div>
  </div>
</div>
```

```

        </a>
        <div class="nav-collapse collapse">
            <ul class="nav">
                <li><a href="/about">درباره</a></li>
                <li><a href="/contact">تماس</a></li>
                <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown"
href="#">Menu
                    ...<span class="caret"></span></a>
                    <ul class="dropdown-menu">
                        <li><a href="http://www.site1.com">site1</a></li>
                        <li><a href="http://www.site2.com">site2</a></li>
                    </ul>
                </li>
            </ul>
            <form class="navbar-search pull-right input-append" action="/search">
                <input class="span2" type="search" />
                <button class="btn" type="button">
جستجو
            </form>
        </div>
    </div>
</nav>
</div>
</div>
</div>

```



navbar نیز همانند بسیاری از کامپوننت‌های Bootstrap، بر اساس ul و liها کار می‌کند. در ابتدای کار، کلاس nav را به تگ ul انتساب می‌دهیم. این مورد با مباحث قسمت قبلی در مورد تدارک برگه‌ها، تفاوتی نمی‌کند. به این ترتیب بولت‌های لیست حذف شده و همچنین تغییر رنگ پس زمینه، با حرکت اشاره‌گر ماوس بر روی آیتم‌ها را سبب می‌شود. سپس کل این ul و liها، داخل تگ nav قرار می‌گیرند با کلاسی مساوی navbar. تا اینجا لیست عمودی تبدیل به یک لیست افقی می‌شود. در ادامه برای اضافه کردن حاشیه‌ای به اطراف این منوی راهنبری و تمایز آن از سایر قسمت‌های صفحه، یک div با کلاس navbar-inner را اضافه خواهیم کرد.

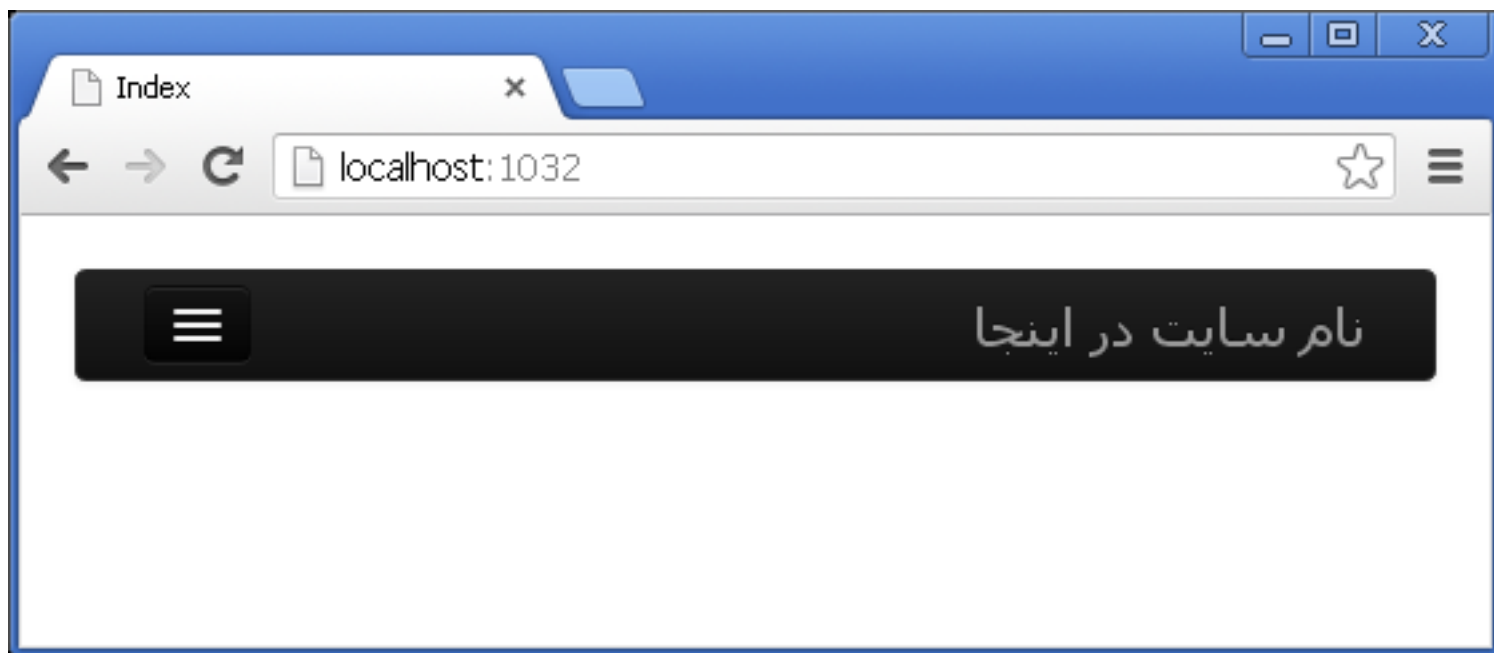
عموماً مرسوم است که در منوی راهنبری اصلی سایت، نام سایت یا متنی مشخص نیز در قسمتی از این منو ظاهر شود. برای انجام اینکار، کافی است یک لینک را با کلاس brand اضافه کنیم. این موارد، اصول کلی تهیه یک منوی راهنبری اصلی سایت را توسط Bootstrap بیان می‌کنند. در ادامه به چند نکته تکمیلی خواهیم پرداخت:

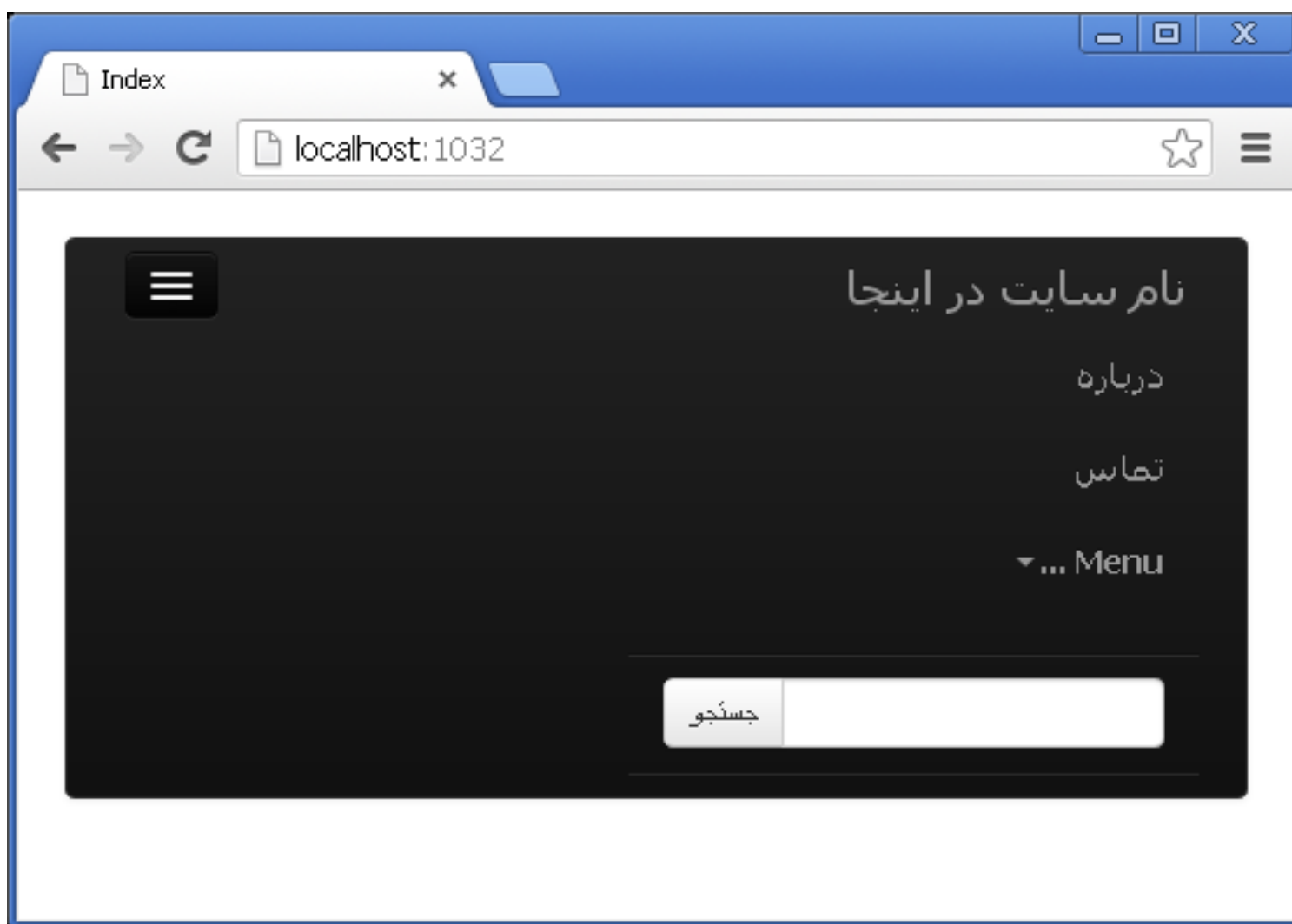
- اگر به تگ nav، کلاس navbar-static-top را اضافه کنیم، فاصله بین منو و اطراف صفحه را حذف می‌کند.
- اگر علاقمند هستیم که منوی بالای صفحه حتی با اسکرول به پایین محتوای نمایش داده شده، همواره نمایان باشد و بر روی

محتوا قرار گیرد، می‌توان به تگ `nav`، کلاس `navbar-fixed-top` را اضافه کرد. یا اگر نیاز است این منو در پایین صفحه به صورت ثابت نمایش داده شود از `navbar-fixed-bottom` می‌شود استفاده کرد.

در این حالت خاص ممکن است منو، قسمتی از ابتدای محتوای صفحه را مخفی کند و این محتوا زیر منو واقع شود. برای بهبود آن باید `margin-top` مرتبط با `body` را مقدار دهی کنید (حدود 40px کافی است).

- استفاده از کلاس `navbar-inverse` (همانند مثال فوق)، سبب معکوس شدن رنگ زمینه منوی راهبری می‌شود.
- همچنین در اینجا نیز می‌توان یک `dropdown-menu` را به نحوی که ملاحظه می‌کنید، به منوی اصلی سایت افزود.
- امکان قرار دادن فرم‌های خاصی مانند فرم جستجو نیز در منوی راهبری وجود دارد. در این حالت از کلاس `navbar-search` مانند مثال فوق استفاده خواهد شد. کلاس `pull-right` این المان را به سمت راست صفحه (در اینجا به سمت چپ با توجه به RTL بودن قالب)، هدایت می‌کند و کلاس `input-append` سبب می‌شود تا دکمه و تکست باکس تعریف شده در یک سطر و کنار هم قرارگیرند تا ظاهر بهتری حاصل شود. در حالت کلی کلاس `navbar-form` نیز برای این نوع فرم‌ها در نظر گرفته شده است.
- اگر دقت کرده باشید، کل محتوای منو، به همراه فرم تعریف شده، در یک `div` با کلاس `nav-collapse` قرار گرفته است. هدف از اینکار ناپدید شدن خودکار قسمت‌های طولانی یا نچندان مهم، در حین مرور سایت با اندازه‌های مرورگر کوچکتر مانند مرورگرهای موبایل است. همچنین اگر بخواهیم در این حالت که قسمت‌هایی ناپدید شده‌اند، یک دکمه با سه سطر که بیانگر منویی مخفی است را نمایش دهیم می‌توان از کلاس `btn-navbar` استفاده کرد؛ که نمونه‌ای از آن را در مثال فوق ملاحظه می‌کنید. این دکمه بر اساس `data-target` مشخص شده، سبب مخفی یا ظاهر شدن قسمتی از صفحه می‌گردد.





تعیین موقعیت کاربر در صفحه به کمک breadcrumbs در Bootstrap

```
<div class="container">
  <div class="row">
    <div class="span12">
      <ul class="breadcrumb">
        <li><a href="/level1">سطح یک</a> <span class="divider"></span></li>
        <li><a href="/level2">سطح دو</a> <span class="divider"></span></li>
        <li class="active">سطح سه</li>
      </ul>
    </div>
  </div>
</div>
```

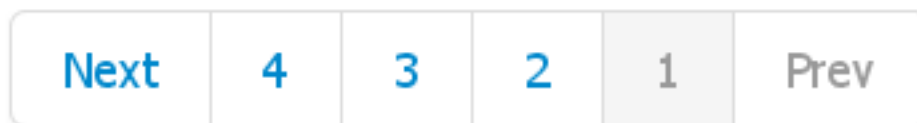
سطح يك / سطح دو / سطح سه

عموما مرسوم است در یک سایت چند سطحی، موقعیت جاری کاربر، به نحوی به او اعلام شود. یکی از این روشها استفاده از breadcrumbs است که نمونه‌ای از آنرا در مثال فوق ملاحظه می‌کنید.

- در اینجا نیز برای تعریف breadcrumbs از ul و li استفاده شده است به همراه کلاس breadcrumb.
- برای ایجاد فاصله و نمایش یک خط جداکننده، از کلاس divider کمک گرفته شده است.
- صفحه جاری با کلاس active مشخص گردیده؛ بدون تعریف لینک.

### تعریف قسمت صفحه بندی یک لیست یا گرید به کمک Bootstrap

```
<div class="container">
  <div class="row">
    <div class="span12">
      <div class="pagination pagination-centered pagination-large">
        <ul>
          <li class="disabled"><a href="#">Prev</a></li>
          <li class="active"><a href="/page/1">1</a></li>
          <li><a href="/page/2">2</a></li>
          <li><a href="/page/3">3</a></li>
          <li><a href="/page/4">4</a></li>
          <li><a href="#">Next</a></li>
        </ul>
      </div>
    </div>
  </div>
</div>
```



برای نمایش زیباتر قسمت صفحه بندی یک لیست یا گرید می‌توان از یک سری ul و li مزین شده با کلاس pagination استفاده کرد. کلاس pagination-centered، آن را در وسط صفحه قرار خواهد داد (حالت‌های چپ و راست نیز وجود دارند). کلاس pagination-large، این قسمت را اندکی بزرگتر از معمول نمایش می‌دهد. در اینجا کلاس disabled، سبب غیرفعال نمایش داده شدن لینک به یک صفحه و کلاس active، بیانگر نمایش صفحه جاری است.

### نمایش برچسب‌ها و اعلانات با زمینه‌های مختلف به کمک Bootstrap

برای نمایش برچسب‌های کوتاه با زمینه‌ای متفاوت از کلاس label استفاده می‌شود. افزودن کلاس‌هایی مانند label-success و label-info، رنگ‌های متفاوتی را سبب خواهند شد. برای گرد کردن بیش از حد گوشه این برچسب‌ها از کلاس badge استفاده می‌شود.

```
<div class="container">
  <div class="row">
    <div class="span12">
      <section>
        <ul class="unstyled">
          <li><span class="label">پیش فرض</span></li>
          <li><span class="label label-success">موفقیت آمیز</span></li>
          <li><span class="label label-warning">خطا</span></li>
          <li><span class="label label-important">مهم است</span></li>
          <li><span class="label label-info">اطلاعات</span></li>
          <li><span class="label label-inverse">رنگ معکوس</span></li>
        </ul>
      </section>
    </div>
  </div>
```

```

        <ul class="unstyled">
            <li><span class="badge">پیش فرض</span></li>
            <li><span class="badge badge-success">موفقیت آمیز</span></li>
            <li><span class="badge badge-warning">اخطار</span></li>
            <li><span class="badge badge-important">مهم است</span></li>
            <li><span class="badge badge-info">اطلاعات</span></li>
            <li><span class="badge badge-inverse">رنگ معکوس</span></li>
        </ul>
    </section>
    <div>
        <div class="alert">
            <button class="close" data-dismiss="alert">&times;</button>
            نمایش اعلانات
        </div>
        <div class="alert alert-danger">
            <button class="close" data-dismiss="alert">&times;</button>
            نمایش اعلانات
        </div>
        <div class="alert alert-info">
            <button class="close" data-dismiss="alert">&times;</button>
            نمایش اعلانات
        </div>
        <div class="alert alert-success">
            <button class="close" data-dismiss="alert">&times;</button>
            نمایش اعلانات
        </div>
        <div class="alert alert-error">
            <button class="close" data-dismiss="alert">&times;</button>
            نمایش اعلانات
        </div>
    </div>
</div>
</div>
</div>

```

پیش فرض

موفقیت آمیز

خطا

مهم است

اطلاعات

رنگ معکوس

پیش فرض

موفقیت آمیز

خطا

مهم است

اطلاعات

رنگ معکوس

×

نمایش اعلانات

×

نمایش اعلانات

×

نمایش اعلانات

×

نمایش اعلانات

×

نمایش اعلانات

همانطور که ملاحظه می‌کنید برای نمایش اعلانی به کاربر می‌توان از کلاس alert استفاده کرد. افزودن کلاس‌هایی مانند alert-danger، رنگ‌های متفاوتی را ارائه خواهند داد و اگر علاقمندید که کاربر بتواند این اخطار را با کلیک بر روی دکمه ضربدر کنار آن حذف کند، از یک دکمه با کلاس close و data-dismiss مساوی alert استفاده کنید. این اطلاعات به صورت خودکار توسط اسکریپت‌های bootstrap پردازش می‌شوند.



## نظرات خوانندگان

نویسنده: راد

تاریخ: ۱۶:۳۹ ۱۳۹۲/۰۴/۲۹

با سلام و تشکر از مطالب مفید سایتتون  
من یک مشکل دارم تمام مراحل رو طبق دستورالعمل شما انجام دادم اما dropdown منوی ایجاد شده وقتی روش کلیک می‌کنم  
هیچ اتفاقی نمی‌افته و باز نمی‌شه.  
CSS ها JS ها رو هم در باندل و هم مستقیماً در layout اضافه کردم باز جواب نگرفتم  
update-package هم هیچ کمکی نکرد لطفا راهنمایی کنید

نویسنده: وحید نصیری

تاریخ: ۱۹:۳۹ ۱۳۹۲/۰۴/۲۹

دریافت یک نمونه مثال: [BootstrapMenu.zip](#)

+ [نحوه بازسازی وابستگی‌ها](#)

یکی دیگر از اجزای تعاملی [Twitter Bootstrap](#) صفحات modal هستند. صفحات modal بر روی صفحه جاری ظاهر شده و کنترل آن‌را در دست می‌گیرند و تا زمانیکه این صفحه ویژه بسته نشود، امکان استفاده از صفحه زیرین، وجود نخواهد داشت. برای استفاده از این امکان ویژه، ابتدا باید یک لینک یا دکمه‌ای، جهت فراخوانی اسکریپت‌های صفحات modal در صفحه تدارک دیده شود.

برای طراحی یک صفحه modal چهار div باید اضافه شوند. بیرونی‌ترین div باید دارای کلاس modal مجموعه Bootstrap باشد. می‌توان به کلاس modal در اینجا کلاس‌های hide fade را هم برای نمونه اضافه کرد. در این حالت، نمایش و بسته شدن صفحه modal به همراه پویانمایی ویژه‌ای خواهد بود.

داخل این div، سه div با کلاس‌های modal-header برای نمایش هدر، modal-body برای نمایش محتوایی در این صفحه modal و modal-footer برای تدارک محتوای footer این صفحه، قرار خواهند گرفت. در این بین هر لینکی با ویژگی data-dismiss=modal، سبب بسته شدن خودکار صفحه باز شده خواهد شد.

### افزونه bootstrapModalConfirm

اگر نکات یاد شده را بخواهیم کپسوله کنیم، می‌توان یک افزونه جدید جی‌کوئری را با نام فایل jquery.bootstrap-modal-confirm.js برای این منظور تدارک دید:

```
// 
(function ($) {
    $.bootstrapModalConfirm = function (options) {
        var defaults = {
            caption: 'تأیید عملیات',
            body: 'آیا عملیات درخواستی اجرا شود؟',
            onConfirm: null,
            confirmText: 'تأیید',
            closeText: 'انصراف',
        };
        var options = $.extend(defaults, options);

        var confirmContainer = "#confirmContainer";
        var html = '&lt;div class="modal hide fade" id="confirmContainer"&gt;' +
            '&lt;div class="modal-header"&gt;' +
            '&lt;a class="close" data-dismiss="modal"&gt;&amp;times;&lt;/a&gt;' +
            '&lt;h5&gt;' + options.caption + '&lt;/h5&gt;&lt;/div&gt;' +
            '&lt;div class="modal-body"&gt;' +
            options.body + '&lt;/div&gt;' +
            '&lt;div class="modal-footer"&gt;' +
            '&lt;a href="#" class="btn btn-success" id="confirmBtn"&gt;' + options.confirmText +
            '&lt;/a&gt;' +
            '&lt;a href="#" class="btn" data-dismiss="modal"&gt;' + options.closeText +
            '&lt;/a&gt;&lt;/div&gt;&lt;/div&gt;';

        $(confirmContainer).remove();
        $(html).appendTo('body');
        $(confirmContainer).modal('show');

        $('#confirmBtn', confirmContainer).click(function () {
            if (options.onConfirm)
                options.onConfirm();
            $(confirmContainer).modal('hide');
        });
    };
})(jQuery);
// ]]&gt;</pre>
</div>
<div data-bbox="62 887 937 942" data-label="Text">
<p>در اینجا به صورت پویا div یک صفحه modal ایجاد و دو دکمه تأیید و لغو به همراه نمایش یک هدر و همچنین محتوایی به کاربر به صفحه اضافه می‌شود. سپس افزونه modal در حالت show، روی این div فراخوانی می‌گردد. در اینجا اگر کاربر بر روی دکمه تأیید کلیک کرد، یک callback به نام onConfirm فراخوانی می‌گردد.</p>
</div>
<div data-bbox="472 956 524 970" data-label="Page-Footer">
<p>۵۰/۱۲۴</p>
</div>
```

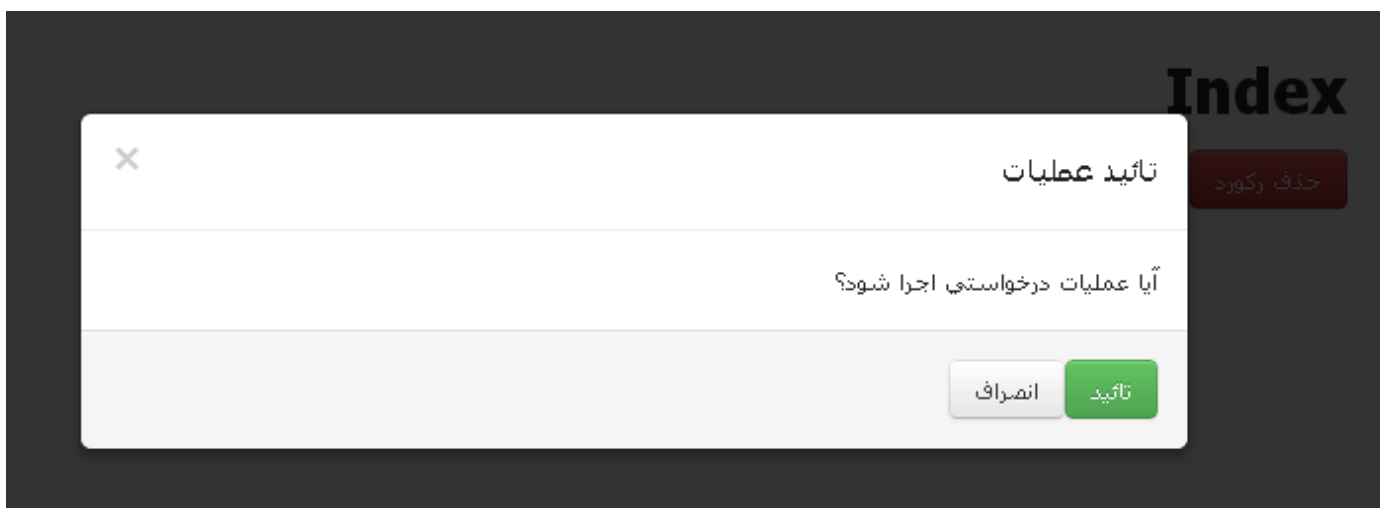
## مثالی از استفاده از افزونه bootstrapModalConfirm

```
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
<a href="#" class="btn btn-danger" id="deleteBtn">حذف رکورد</a>
@section JavaScript
{
    <script type="text/javascript">
        $(function () {
            $("#deleteBtn").click(function (e) {
                e.preventDefault(); // عمل نکند
                می‌خواهیم لینک به صورت معمول عمل نکند
                $.bootstrapModalConfirm({
                    caption: 'تأیید عملیات',
                    body: 'آیا عملیات درخواستی اجرا شود؟',
                    onConfirm: function () {
                        alert('در حال انجام عملیات');
                    },
                    confirmText: 'تأیید',
                    closeText: 'انصراف'
                });
            });
        });
    </script>
}
```

در مثال فوق، اگر کاربر بر روی لینک حذف رکورد، که به صورت یک دکمه مزین شده است کلیک کند، در صورت تأیید، قسمت onConfirm اجرا خواهد شد.

# Index

حذف رکورد



## نظرات خوانندگان

نویسنده: کیارش سلیمان زاده  
تاریخ: ۲۳:۵۵ ۱۳۹۲/۰۳/۲۴

البته به نظر من ، راه بهتر برای اینکار استفاده از افرونه [bootbox](#) هستش.

نویسنده: وحید نصیری  
تاریخ: ۱۲:۵۴ ۱۳۹۲/۰۳/۲۵

و برای اطلاع رسانی توسط alertها به صورت پویا:

[Twitter Bootstrap Extensions: Notifications](#)

[jQuery Bootstrap Alerts](#)

نویسنده: رضا  
تاریخ: ۱۲:۴۵ ۱۳۹۳/۰۲/۳۱

به چه صورت می‌تونیم تو یه لیست ازش استفاده کنیم ..یعنی هریک از این حذف‌ها تو یه سطر باشن

نویسنده: وحید نصیری  
تاریخ: ۱۳:۰۱ ۱۳۹۳/۰۲/۳۱

مطلب + نظرات « [حذف یک ردیف از اطلاعات به همراه پویانمایی محو شدن اطلاعات آن توسط jQuery در ASP.NET MVC](#) »

نویسنده: محمدرضا2  
تاریخ: ۱۵:۵۶ ۱۳۹۳/۰۶/۰۲

سلام.

چطور می‌توان سایز Modal را Dynamic و بر اساس محتوا تنظیم کرد.

مثلا در یک گالری تصاویر، عکسها میتوانند در سایزهای مختلف در سایت نمایش داده شوند؟

نویسنده: وحید نصیری  
تاریخ: ۱۸:۴۴ ۱۳۹۳/۰۶/۰۲

نیاز به یک سری تغییرات در CSS و JS دارد:

```
.modal.modal-wide .modal-dialog {
  width: 90%;
}
.modal-wide .modal-body {
  overflow-y: auto;
}

/* irrelevant styling */
body { text-align: center; }
body p {
  max-width: 400px;
  margin: 20px auto;
}
#tallModal .modal-body p { margin-bottom: 900px }
```

```
// when .modal-wide opened, set content-body height based on browser
// height; 200 is appx height of modal padding, modal title and button bar

$(".modal-wide").on("show.bs.modal", function() {
  var height = $(window).height() - 200;
  $(this).find(".modal-body").css("max-height", height);
});
```

استفاده از modal dialogs مجموعه Twitter Bootstrap برای گرفتن تأیید از کاربر

---

```
});
```

[مثال قابل اجرا](#)

+ یک مثال دیگر: [^](#)

اصول نمایش اطلاعات مودال به کمک bootstrap در مطلب « [استفاده از modal dialogs مجموعه Twitter Bootstrap برای گرفتن تأیید از کاربر](#) » بررسی شدند.

در این قسمت قصد داریم یک فرم Ajax ایی را در ASP.NET MVC به همراه تمام مسایل اعتبارسنجی، پردازش اطلاعات و غیره را به کمک Twitter Bootstrap و jQuery Ajax پیاده سازی کنیم.

### تهیه افزونه jquery.bootstrap-modal-ajax-form.js

از این جهت که مباحث مرتبط با نمایش و پردازش فرم‌های مودال Ajax ایی به کمک Twitter Bootstrap اندکی نکته دار و طولانی هستند، بهتر است این موارد را به شکل یک افزونه، کپسوله کنیم. کدهای کامل افزونه jquery.bootstrap-modal-ajax-form.js را ادامه ملاحظه می‌کنید:

```
// 
(function ($) {
    $.bootstrapModalAjaxForm = function (options) {
        var defaults = {
            renderModalPartialViewUrl: null,
            renderModalPartialViewData: null,
            postUrl: '/',
            loginUrl: '/login',
            beforePostHandler: null,
            completeHandler: null,
            errorHandler: null
        };
        var options = $.extend(defaults, options);

        var validateForm = function (form) {
            // فعال سازی دستی اعتبار سنجی جی کوئری
            var val = form.validate();
            val.form();
            return val.valid();
        };

        var enableBootstrapStyleValidation = function () {
            $.validator.setDefaults({
                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');
                }
            });
        };

        var enablePostBackValidation = function () {
            $('form').each(function () {
                $(this).find('div.control-group').each(function () {
                    if ($(this).find('span.field-validation-error').length &gt; 0) {
                        $(this).addClass('error');
                    }
                });
            });
        };
    };
});
// ]&gt;</pre>
</div>
<div data-bbox="470 955 524 970" data-label="Page-Footer">۵۵/۱۲۴</div>
```

```

var processAjaxForm = function (dialog) {
    $('form', dialog).submit(function (e) {
        e.preventDefault();

        if (!validateForm($(this))) {
            //اگر فرم اعتبار سنجی نشده، اطلاعات آن ارسال نشود
            return false;
        }

        //در اینجا می‌توان مثلا دکمه‌ای را غیرفعال کرد
        if (options.beforePostHandler)
            options.beforePostHandler();

        //اطلاعات نباید کش شوند
        $.ajaxSetup({ cache: false });
        $.ajax({
            url: options.postUrl,
            type: "POST",
            data: $(this).serialize(),
            success: function (result) {
                if (result.success) {
                    $('#dialogDiv').modal('hide');
                    if (options.completeHandler)
                        options.completeHandler();
                } else {
                    $('#dialogContent').html(result);
                    if (options.errorHandler)
                        options.errorHandler();
                }
            }
        });
        return false;
    });
};

var mainContainer = "<div id='dialogDiv' class='modal hide fade in'><div
id='dialogContent'></div></div>";
enableBootstrapStyleValidation(); //با آن جهت اعتبارسنجی یکپارچه
$.ajaxSetup({ cache: false });
$.ajax({
    type: "POST",
    url: options.renderModalPartialViewUrl,
    data: options.renderModalPartialViewData,
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    complete: function (xhr, status) {
        var data = xhr.responseText;
        if (xhr.status == 403) {
            window.location = options.loginUrl; //در حالت لاگین نبودن شخص اجرا می‌شود
        }
        else if (status === 'error' || !data) {
            if (options.errorHandler)
                options.errorHandler();
        }
        else {
            var dialogContainer = "#dialogDiv";
            $(dialogContainer).remove();
            $(mainContainer).appendTo('body');

            $('#dialogContent').html(data); //دریافت پویای اطلاعات مودال دیالوگ
            $.validator.unobtrusive.parse("#dialogContent"); //فعال سازی اعتبارسنجی فرمی که با
            //ایجکس بارگذاری شده
            enablePostBackValidation();
            //و سپس نمایش آن به صورت مودال
            $('#dialogDiv').modal({
                backdrop: 'static', //با کلیک کاربر روی صفحه، صفحه مودال بسته نمی‌شود
                keyboard: true
            }, 'show');
            //تحت نظر قرار دادن این فرم اضافه شده
            processAjaxForm('#dialogContent');
        }
    }
});
})(jQuery);
// ]]>

```



- توابع `enablePostbackValidation` و `enableBootstrapStyleValidation` در مطلب « [اعمال کلاس‌های ویژه اعتبارسنجی](#) » [Twitter bootstrap به فرم‌های ASP.NET MVC](#) بررسی شدند.

- این افزونه با توجه به مقدار `renderModalPartialViewUrl`، یک `partial view` را از برنامه ASP.NET MVC درخواست می‌کند.  
 - سپس این `partial view` را به صورت خودکار به صفحه اضافه کرده و آن‌را به صورت `modal` نمایش می‌دهد.  
 - پس از افزودن فرم Ajax ایی دریافتی، مسایل اعتبارسنجی را به آن اعمال کرده و سپس دکمه `submit` آن‌را تحت کنترل قرار می‌دهد.

- در زمان `submit`، ابتدا بررسی می‌کند که آیا فرم معتبر است و اعتبارسنجی آن بدون مشکل است؟ اگر اینچنین است، اطلاعات فرم را به آدرس `postUrl` به صورت Ajax ایی ارسال می‌کند.

### کدهای مدل برنامه

```
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;

namespace Mvc4TwitterBootstrapTest.Models
{
    public class User
    {
        public int Id { set; get; }

        [DisplayName("نام")]
        [Required(ErrorMessage="لطفا نام را تکمیل کنید")]
        public string Name { set; get; }

        [DisplayName("نام خانوادگی")]
        [Required(ErrorMessage = "لطفا نام خانوادگی را تکمیل کنید")]
        public string LastName { set; get; }
    }
}
```

در اینجا یک مدل ساده‌را به همراه ویژگی‌های اعتبارسنجی و نام‌های نمایشی خواص ملاحظه می‌کنید.

### کدهای کنترلر برنامه

```
using System.Web.Mvc;
using Mvc4TwitterBootstrapTest.Models;

namespace Mvc4TwitterBootstrapTest.Controllers
{
    public class ModalFormAjaxController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View(); // نمایش صفحه اولیه
        }

        [HttpPost] // برای این حالت امن‌تر است
        //[AjaxOnly]
        public ActionResult RenderModalPartialView()
        {
            // رندر پارشال ویوو صفحه مودال به همراه اطلاعات مورد نیاز آن
            return PartialView(viewName: "_ModalPartialView", model: new User { Name = "", LastName = "" });
        }

        [HttpPost]
        //[AjaxOnly]
        public ActionResult Index(User user) // ذخیره سازی اطلاعات
        {
            if (this.ModelState.IsValid)
            {
                //todo: SaveChanges;
                return Json(new { success = true });
            }

            this.ModelState.AddModelError("", "خطایی رخ داده است");
        }
    }
}
```

```

        return PartialView("_ModalPartialView", user);
    }
}

```

کدهای کنترلر برنامه در این حالت از سه قسمت تشکیل می‌شود:  
 الف) متد Index حالت HttpGet که صفحه ابتدایی را نمایش خواهد داد.  
 ب) متد RenderModalPartialView یک partial view اضافه شده به برنامه به نام ModalPartialView.cshtml را بازگشت می‌دهد. این partial view در حقیقت همان فرمی است که قرار است به صورت مودال نمایش داده شود و پردازش آن نیز Ajax ایی باشد.  
 ج) متد Index حالت HttpPost که نهایتاً اطلاعات فرم مودال را دریافت خواهد کرد. اگر پردازش موفقیت آمیز بود، نیاز است همانند کدهای فوق return Json صورت گیرد. در غیراینصورت مجدداً همان partial view را بازگشت دهید.

### کدهای Index.cshtml

```

@{
    ViewBag.Title = "Index";
    var renderModalPartialViewUrl = Url.Action("RenderModalPartialView", "ModalFormAjax");
    var postDataUrl = Url.Action("Index", "ModalFormAjax");
}
<h2>
    Index</h2>
<a href="#" class="btn btn-primary" id="btnCreate">ثبت اطلاعات</a>
@section JavaScript
{
    <script type="text/javascript">
        $(function () {
            $('#btnCreate').click(function (e) {
                e.preventDefault(); // صورت معمول عمل نکند
                $.bootstrapModalAjaxForm({
                    postUrl: '@postDataUrl',
                    renderModalPartialViewUrl: '@renderModalPartialViewUrl',
                    renderModalPartialViewData: {},
                    loginUrl: '/login',
                    beforePostHandler: function () {
                    },
                    completeHandler: function () {
                        // Refresh: صفحه زیرین باشد
                        // location.reload();
                    },
                    errorHandler: function () {
                    }
                });
            });
        });
    </script>
}

```

این کدها متناظر هستند با کدهای view اکشن متد Index در حالت Get.  
 - در اینجا یک لینک ساده در صفحه قرار گرفته و به کمک کلاس btn مجموعه bootstrap به شکل یک دکمه مزین شده است.  
 - در ادامه نحوه استفاده از افزونه‌ای را که در ابتدای بحث طراحی کردیم، ملاحظه می‌کنید. کار با آن بسیار ساده است و تنها باید مسیرهای ارسال اطلاعات نهایی به سرور یا postDataUrl و مسیر دریافت partial view رندر شده یا renderModalPartialViewUrl به آن معرفی شود. سایر مسایل آن خودکار است.

### کدهای ModalPartialView.cshtml یا همان فرم مودال برنامه

```

@model Mvc4TwitterBootstrapTest.Models.User
<div class="modal-header">
    <button type="button" class="close" data-dismiss="modal" aria-hidden="true">

```

```

        &times;</button>
    <h5>
        افزودن کاربر جدید
    </h5>
</div>
@using (Html.BeginForm("Index", " ModalFormAjax", FormMethod.Post, new { @class = "modal-form" }))
{
    <div class="modal-body">
        @Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })
        <fieldset class="form-horizontal">
            <legend>مشخصات کاربر</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>
        </fieldset>
    </div>

    <div class="modal-footer">
        <button class="btn btn-primary" type="submit">
            ارسال
        </button>
        <button class="btn" data-dismiss="modal" aria-hidden="true">
            انصراف
        </button>
    </div>
}

```

در اینجا [اطلاعات فرمی را](#) ملاحظه می‌کنید که قرار است به صورت مودال نمایش داده شود. نحوه طراحی آن بر اساس [نکات form-horizontal](#) است. همچنین div‌های modal-header، modal-body و modal-footer نیز به این فرم ویژه اضافه شده‌اند تا به خوبی توسط bootstrap پردازش گردد. حاصل نهایی این مبحث را در دو شکل ذیل ملاحظه می‌کنید. صفحه index نمایش دهنده یک دکمه و در ادامه باز شدن یک فرم مودال، پس از کلیک بر روی دکمه ثبت اطلاعات.

# Index

ثبت اطلاعات

# Index

ثبت اطلاعات

×

افزودن کاربر جدید

مشخصات کاربر

نام

لطفا نام را تکمیل کنید

نام خانوادگی

لطفا نام خانوادگی را تکمیل کنید

انصراف

ارسال

## نظرات خوانندگان

نویسنده: وحید نصیری  
تاریخ: ۱۴:۵۶ ۱۳۹۲/۰۴/۰۵

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

+ [نحوه بازسازی وابستگی‌ها](#)

نویسنده: دانشجو  
تاریخ: ۲۳:۲۳ ۱۳۹۲/۰۴/۱۸

عرض پاپ آپ چطور تعیین میشه؟  
امکان تمام صفحه شدن آن وجود دارد؟

نویسنده: وحید نصیری  
تاریخ: ۲۳:۳۷ ۱۳۹۲/۰۴/۱۸

با css قابل تنظیم است. برای تنظیم عرض و سایر مشخصات تمام صفحات مودال:

```
body .modal {  
    width: 560px;  
    margin-left: -280px;  
}
```

و یا در مثال فوق برای یک صفحه خاص:

```
$('#dialogDiv').modal({  
    backdrop: 'static', // صفحه مودال بسته نمی‌شود  
    keyboard: true  
}, 'show').css({  
    'width': '100px',  
    'max-width': '1000px',  
    'margin-left': '10px' });
```

نویسنده: علی رضا امیدیان  
تاریخ: ۱۸:۲۹ ۱۳۹۲/۰۴/۲۷

با سلام  
در اکشن زیر

```
public ActionResult Index(User user) // ذخیره سازی اطلاعات  
{  
    if(this.ModelState.IsValid)  
    {  
        //todo: SaveChanges;  
        return Json(new{ success = true});  
    }  
  
    this.ModelState.AddModelError("", "خطایی رخ داده است");  
    return PartialView("_ModalPartialView", user);  
}
```

اگر 4 خط اول را کامنت کنیم، یعنی فرض کنیم خطایی اتفاق می‌افتد، که در برنامه‌های واقعی امکان پذیر است، از بار دوم به بعد، فرم از حالت مودال خارج می‌شود. علت چیست و راه حل کدام است؟

نویسنده: وحید نصیری  
تاریخ: ۱۹:۵۸ ۱۳۹۲/۰۴/۲۷

در متد processAjaxForm در قسمت options.errorHandler آن چون dialogContent مجدداً مقدار دهی می‌شود، نیاز است سیم کشی‌های آن مجدداً برقرار شوند، شامل مباحث اعتبارسنجی و همچنین تحت کنترل قرار دادن دکمه submit آن. یعنی به صورت زیر

```
} else {
    $('#dialogContent').html(result);

    $.validator.unobtrusive.parse("#dialogContent");
    enablePostbackValidation();
    processAjaxForm('#dialogContent');

    if (options.errorHandler)
        options.errorHandler();
}
```

نویسنده: علی رضا امیدیان  
تاریخ: ۱۴:۳۷ ۱۳۹۲/۰۴/۲۸

با سلام و تشکر از پاسخ شما

نویسنده: پیام  
تاریخ: ۸:۵۵ ۱۳۹۲/۰۶/۰۱

با سلام آقای نصیری من برای استفاده از این مثال تو سایت خودم مجبور از بسته jquery-1.9.1 استفاده کنم این درحالی که من برای اجرای منوی superfish باید از بسته ای jquery-1.6 به پایین استفاده کنم برای حل همچنین مشکلاتی چه کاری باید کنم ؟

نویسنده: وحید نصیری  
تاریخ: ۹:۰۰ ۱۳۹۲/۰۶/۰۱

- از آخرین نسخه افزونه یاد شده استفاده کنید. اگر ندارد، با نویسنده آن تماس بگیرید و درخواست ارتقاء آن را بدهید.  
- jquery-1.9.1 نباید تداخلی با کار شما داشته باشد. یک سری از مسایل از نگارش 2 آن به بعد هست که [حذف شدند](#) و منسوخ شده اعلام گردیدند. برای رفع آن ، نیاز است از کتابخانه [jQuery Migrate](#) نیز استفاده کنید.

نویسنده: سیروان عقیفی  
تاریخ: ۲۰:۳۴ ۱۳۹۲/۰۷/۱۰

ممنون از شما واقعاً مطلب جالبی بود،

فقط یک سوال :

کلاس‌های مدل من Product و Category است، نحوه نمایش گروه محصولات در DropDownList به چه صورتی است؟ ابتدا من یک ViewModel ایجاد کردم :

```
public class ProductCategoryViewModel
{
    public Product Product{get;set;}
    public Category Category{get;set;}
}
```

و سپس در RenderModalPartialView به PartialView مربوطه پاس دادم :

```
public ActionResult RenderModalPartialView()
{
    //رندر پارشال ویوو صفحه مودال به همراه اطلاعات مورد نیاز آن
    return PartialView(viewName: "_ModalPartialView", model: new ProductCategoryViewModel());
}
```

در PartialView هم برای نمایش نام گروه محصولات از طریق ViewBag لیست گروه محصولات داده شده را در یک

DropDownList به صورت زیر استفاده کردم :

```
@Html.DropDownListFor(model => model.Product.CategoryId, (SelectList)ViewBag.CategryNames, "--گروه--")
(موردنظر را انتخاب کنید)
```

اما روش فوق کارساز نیست و با کلیک بر روی باتن Modal را نمایش نمی‌دهد .

و به سوال دیگه اینکه چطور قالب [T4 تولید شده](#) رو برای حالت فرم‌های مودال بازنویسی کنیم؟ البته ظاهراً باید توی قسمت :

```
if(mvcHost.IsPartialView) {
#>
<#
```

قالب موردنظر ( ModalPartialView\_ ) برای حالت مودال فرم رو باید بنویسم، من با سینتکس T4 آشنا نیستم ، میشه لطف کنید راهنمایی کنید؟

نویسنده: وحید نصیری  
تاریخ: ۲۱:۵۳ ۱۳۹۲/۰۷/۱۰

- این مطلب برای بوت استرپ 2 نوشته شده. (از فایل‌های نگارش 3 استفاده نکنید)  
+ چه چیزی رو نمایش نمی‌ده؟ کلا صفحه باز نمیشه؟ یا اینکه صفحه باز میشه اما دراپ داون آن خالی است؟ اگر صفحه باز نمیشه که سورت کامل این قسمت [در اولین نظر بحث](#) ارسال شده. مقایسه کنید چه مواردی رو لحاظ نکردید. همچنین برنامه رو [در فایرباگ دیباگ کنید](#) شاید اسکریپتی فراموش شده. اگر دراپ داون صفحه باز شده خالی است، طبیعی هست. چون در متد RenderModalPartialView که نوشتید ViewBag ایی مقدار دهی نشده. بنابراین دیتاسورت مدنظر شما نال هست. روش بهتر این است که یک خاصیت به ViewModel ایی که تعریف کردید اضافه کنید:

```
public IEnumerable<SelectListItem> Names { get; set; }
```

بعد این خاصیت را در زمان بازگشت اطلاعات از اکشن متد مقدار دهی کنید:

```
model: new ProductCategoryViewModel { Names = ... }
```

- نیازی نیست کلا بازنویسی شود. در Razor view engine تفاوتی بین فایل نهایی View و Partial View نیست. به همین جهت برای اینکه این‌دو با هم اشتباه گرفته نشوند یک \_ در ابتدای نام partial view قرار می‌دهند. فقط چند div مخصوص modal-header و موارد دیگری که در متن ذکر شده را باید به قالب قبلی اضافه کنید تا برای فرم‌های مودال هم کار کند. تفاوت دیگری ندارد.

نویسنده: سیروان عقیفی  
تاریخ: ۲۱:۵۵ ۱۳۹۲/۰۷/۱۰

تغییر کدهای T4 برای حالت فرم‌های Modal به صورت زیر است :

```
<#
  if (!mvcHost.IsContentPage) {
#>

<#
  }
#>
<div class="modal-header">
  <button type="button" class="close" data-dismiss="modal" aria-hidden="true">
    &times;</button>
```

```

    <#= mvcHost.ViewDataType.Name #>
</div>
@using (Html.BeginForm()) {
<div class="modal-body">
    @Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })

    <fieldset class="form-horizontal">
        <legend><#= mvcHost.ViewDataType.Name #></legend>

<#
foreach (ModelProperty property in GetModelProperties(mvcHost.ViewDataType)) {
    if (!property.IsPrimaryKey && !property.IsReadOnly && property.Scaffold) {
#>
        <div class="control-group">
<#
            if (property.IsForeignKey) {
#>
                @Html.LabelFor(model => model.<#= property.Name #>, "<#= property.AssociationName #>", new
{@class="control-label"})
<#
            } else {
#>
                @Html.LabelFor(model => model.<#= property.Name #>, new {@class="control-label"})
<#
            }
#>

            <div class="controls">
<#
                if (property.IsForeignKey) {
#>
                    @Html.DropDownList("<#= property.Name #>", String.Empty)
<#
                } else {
#>
                    @Html.EditorFor(model => model.<#= property.Name #>)
<#
                }
#>
                @Html.ValidationMessageFor(model => model.<#= property.Name #>, null, new {@class="help-
inline"})
</div>
            </div>

<#
        }
#>
    </fieldset>
</div>

<div class="modal-footer">
    <button class="btn btn-primary" type="submit">
        ارسال
    <button class="btn" data-dismiss="modal" aria-hidden="true">
        انصراف
    </div>
}

```

حالا تنها مشکلم نحوه نمایش گروه محصولات داخل DropDownList است.

نویسنده: مسعود بهرامی  
تاریخ: ۵:۳۸ ۱۳۹۲/۰۹/۰۲

حال یه سوال اگه بخواهیم مثلا پس از اینکه کاربر اطلاعات را به شکل درست وارد کرده و اطلاعات هم به درستی ذخیره شده باشه یعنی تمام کار به درستی انجام شده باشه پس از بستن پنجره مودال یه پیام مثلا به شکل پاپ آپ به کاربر نشون بدیم مبنی بر صحت انجام کار باید به چه صورت عمل کرد یا مثل بالا جایی که شما نوشتین خطایی رخ داده است. منظورم شبیه چیزیه که تو همین وبلاگ عمل میشه یعنی اینجا پس از مثلا نوشتن این نظر و ارسال بدون اشکال به سرور به بنده یه پیام نشون میده که بدونم نتیجه‌ی کارمو با تشکر (:



نویسنده: وحید نصیری  
تاریخ: ۹:۲۶ ۱۳۹۲/۰۹/۰۲

این مورد پیش بینی شده در این مثال. اگر به خروجی اکشن متد دقت کنید، چنین چیزی است:

```
return Json(new { success = true });
```

همین مورد یعنی بررسی مقدار success دریافتی، جایی که عملیات Ajax ایی ارسال اطلاعات پایان می‌یابد انجام شده:

```
if (result.success) {
    $('#dialogDiv').modal('hide');
    if (options.completeHandler)
        options.completeHandler();
}
```

بنابراین تنها کاری که باید انجام بدید، قرار دادن کدهای نمایش اطلاعات نهایی در callbackهای completeHandler و با errorHandler مربوط به افزونه \$.bootstrapModalAjaxForm است. اینجا دست شما باز است. اگر علاقمند بودید از یک alert ساده استفاده کنید یا از همین روش نمایش صفحات مودال [به نحوی دیگر](#) یا از صدها [افزونه jQuery](#) موجود برای نمایش پیام‌ها.

نویسنده: مسعود بهرامی  
تاریخ: ۱۶:۵۰ ۱۳۹۲/۰۹/۰۲

مرسی؛ به مورد دیگه با عرض پوزش تو این فرم مودال چون Ajax هستش به مشکل که داریم وقتی که بخواهیم دیتا سورس رو مثلاً برای کومبو باکس تلریک پر کنیم مثل مثال ساده‌ی زیر

```
@(Html.Kendo().ComboBoxFor(model => model.ParentId)
    .DataTextField("Name")
    .DataValueField("Id")
    .HtmlAttributes(new { style = "width:150px" })
    .Filter("contains")
    .AutoBind(false)
    .MinLength(3)
    .Placeholder(CategoryResource.Category)
    .DataSource(source =>
    {
        source.Read(read =>
        {
            read.Action("GetAllGroups", "Utility");
        })
        .ServerFiltering(false);
    })
)
```

چون تو این حالت اصلاً کومبو کندو رندر نمیشه برای این حالت که مشکل اساسی هم هستش آیا راهی وجود داره من اطلاعات رو تو ViewBag هم ریختم و تو کد بالا بجای دیتا سورس از خاصیت بایند اون بصورت دستی استفاده کردم یعنی اونو دستی بایند کردم به ViewBag اما باز هم قایده نداشت چون کومبو باکس کندو اصلاً Render نمیشه ممنون میشم اگه راهنمایی کنین :

نویسنده: وحید نصیری  
تاریخ: ۱۹:۲ ۱۳۹۲/۰۹/۰۲

[کمی بالاتر](#) در جواب سؤال مشابهی پاسخ دادم. باید یک ViewModel درست کنید جایی که return PartialView توسط عملیات Ajax ایی درخواست می‌شود، منبع داده View در حال رندر را تامین کند.

نویسنده: مجید  
تاریخ: ۲۲:۰ ۱۳۹۲/۰۹/۱۴

اگه بخواهیم از بوت استرپ 3 استفاده کنیم چه تغییری رو باید انجام بدیم؟

نویسنده: وحید نصیری  
تاریخ: ۲۳:۲۸ ۱۳۹۲/۰۹/۱۴

در دوره « [ارتقاء به Bootstrap 3](#) » بحث شده.

نویسنده: حسین  
تاریخ: ۱۷:۳۸ ۱۳۹۲/۱۱/۲۵

با سلام و تشکر  
موقعی که اطلاعات اینجوری باشه درست کار میکنه و نتیجه رو نمایش میده

```
return Json( "test");
```

```
completeHandler: function (data) {  
    alert(data)  
}
```

ولی موقعی که مثلا اینجوری مینویسم

```
return Json(  
    new  
    {  
        success = true,  
        message = Resources.Global.Texts.Successful  
    });
```

```
completeHandler: function (data) {  
    alert(data.success)  
}
```

خطای مربوطه هم Cannot read property 'message' of undefined میده تو فایر باگ هم میبینم اطلاعات در response وجود داره ولی اینجا نمایش نمیده

نویسنده: وحید نصیری  
تاریخ: ۱۷:۴۷ ۱۳۹۲/۱۱/۲۵

پروژه سورس باز هست. به قسمت success سورس آن دقت کنید (سطر 74 کد افزونه):

```
success: function (result) {  
    if (result.success) {  
        $('#dialogDiv').modal('hide');  
        if (options.completeHandler)  
            options.completeHandler();  
    }  
}
```

به completeHandler فراخوانی شده، اطلاعاتی ارسال نمی‌شود.  
مطابق نیاز خودتان این قسمت را تغییر داده و استفاده کنید. مثلا result دریافتی را به آن متد ارسال کنید.

نویسنده: مجتبی فخاری  
تاریخ: ۲۳:۳۴ ۱۳۹۲/۱۲/۲۰

با سلام  
منظورتون رو از مطلب زیر نفهمیدم. میشه بگید چطوری میتونیم با استفاده از alert خود بوت استرپ نتیجه را از کنترلر به فرم مدال ارسال کنیم و در آنجا نمایش دهیم؟ «این مورد پیش بینی شده در این مثال. اگر به خروجی اکشن متد دقت کنید، چنین چیزی است:

return Json( new { success = true } );	1
--	---

همین مورد یعنی بررسی مقدار success دریافتی، جایی که عملیات Ajax ایی ارسال اطلاعات پایان می‌یابد انجام شده:

if (result.success) {	1
\$( '#dialogDiv' ).modal( 'hide' );	2
if (options.completeHandler)	3
options.completeHandler();	4

بنابراین تنها کاری که باید انجام بدید، قرار دادن کدهای نمایش اطلاعات نهایی در callback های completeHandler و یا errorHandler مربوط به افزونه \$.bootstrapModalAjaxForm است. اینجا دست شما باز است. اگر علاقمند بودید از یک alert ساده استفاده کنید»

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۱۲/۲۰ ۲۳:۵۸

من اینجا یک خاصیت دلخواه به نام success اضافه کرده‌ام. دست شما باز است. تعداد خاصیت‌های بیشتری را به شیء anonymous بازگشت داده شده از کنترلر اضافه کنید. محدودیتی ندارد. این شیء نهایتاً در سمت کلاینت تحت عنوان result دریافت شده. یا مستقیماً با آن کار کنید یا این result را به متد completeHandler ارسال کنید تا در جای دیگری توسط completeHandler تعریف شده قابل استفاده باشد.

نویسنده: ع طاهری  
تاریخ: ۱۳۹۲/۱۲/۲۵ ۱۴:۰۶

سلام.  
برای پیغامهایی که در این سایت نمایش داده می‌شوند مانند "پیغامهای خطا" به رنگ قرمز یا "پیغامهای info" به رنگ آبی و ... که در وسط صفحه ظاهر شده و بعد از چند ثانیه ناپدید می‌شوند از modal استفاده شد؟ لطفاً راهنمایی کنید که چگونه میتوانم همچنین پیغامهایی را نمایش دهم.  
با سپاس فراوان

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۱۲/۲۵ ۱۴:۲۴

از [noty](#) استفاده شده.

نویسنده: مجتبی فخاری  
تاریخ: ۱۳۹۳/۰۲/۰۱ ۰:۳۵

با سلام  
اگر بخواهیم برای Edit از این مدل استفاده کنیم چطوری می‌توانیم اطلاعات مثلاً Id را به متد مورد نظر پاس بدهیم. آیا باید در این خط Id, renderModalPartialViewData: {}, ارسال نماییم؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۲/۰۱ ۰:۴۴

```
$.ajax({
    type: "POST",
    url: options.renderModalPartialViewUrl,
```

```
data: options.renderModalPartialViewData,
```

مطابق سورس فوق، data ارسالی عملیات ajax با renderModalPartialViewData مقدار دهی می‌شود.

نویسنده: M.Shakeri

تاریخ: ۱۴:۵۷ ۱۳۹۳/۰۵/۲۵

تو مثالی که آپلود کردین اگه پارامتر بفرستیم خطای 500 میده.

```
$.bootstrapModalAjaxForm({
    postUrl: '@postDataUrl',
    renderModalPartialViewUrl: '@renderModalPartialViewUrl',
    renderModalPartialViewData: { id: 1},
```

```
public ActionResult RenderModalPartialView(int id)
{
    //رندر پارشال ویوو صفحه مودال به همراه اطلاعات مورد نیاز آن
    return PartialView(viewName: "_ModalPartialView", model: new User { Name = "", LastName = 
"" });
}
```

❌ POST <http://localhost:1032/ModalFormAjax/RenderModalPartialView> 500 (Internal Server Error)

نویسنده: وحید نصیری

تاریخ: ۱۵:۳۶ ۱۳۹۳/۰۵/۲۵

```
renderModalPartialViewData: JSON.stringify({ id: 1 }),
```

## گروه بندی دکمه‌ها در Twitter bootstrap

```
<div class="btn-group" data-toggle="buttons-radio">
  <button class="btn" type="button">بلی</button>
  <button class="btn" type="button">خیر</button>
</div>
```

در این مثال دو دکمه را ملاحظه می‌کنید که در یک div با کلاس btn-group محصور شده‌اند. به این ترتیب این دو دکمه در کنار هم، همانند دکمه‌های یک toolbar قرار خواهند گرفت. همچنین در بوت‌استرپ امکان انتساب ویژگی data-toggle=buttons-radio نیز به این div وجود دارد. در این حالت، این دکمه‌ها همانند دکمه‌های رادیویی رفتار خواهند کرد:



در ادامه قصد داریم یک Editor template و یک Display template مخصوص را جهت تدارک یک چنین دکمه‌هایی، برای مدیریت خواص Boolean ایجاد کنیم. به عبارتی اگر مدل برنامه چنین تعاریفی را داشت:

```
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;

namespace Mvc4TwitterBootstrapTest.Models
{
    public class User
    {
        public int Id { set; get; }

        [DisplayName("نام")]
        [Required(ErrorMessage="تکمیل کنید")]
        public string Name { set; get; }

        [DisplayName("نام خانوادگی")]
        [Required(ErrorMessage = "تکمیل کنید")]
        public string LastName { set; get; }

        [DisplayName("فعال است؟")]
        [UIHint("BootstrapBoolean")]
        public bool? IsActive { set; get; }
    }
}
```

فیلد nullable bool آن به صورت خودکار به شکل زیر رندر شود:

## مشخصات کاربر

نام

نام خانوادگی

نامشخص

خیر

بلی

فعال است؟

ارسال

### تهیه قالب ادیتور Views\Shared\EditorTemplates\BootstrapBoolean.cshtml

```
@model bool?
@{
    var yesIsSelected = Model.HasValue && Model.Value ? "active" : null;
    var noIsSelected = Model.HasValue && !Model.Value ? "active" : null;
    var isIndeterminate = !Model.HasValue ? "active" : null;
    var htmlField = ViewData.TemplateInfo.HtmlFieldPrefix;
}
@Html.HiddenFor(model => model)
<div class="btn-group" data-toggle="buttons-radio">
    <button type="button" class="btn btn-info @yesIsSelected bool-@htmlField"
    onclick="$('#@htmlField').val(true);">
        بلی
    </button>
    <button type="button" class="btn btn-info @noIsSelected bool-@htmlField"
    onclick="$('#@htmlField').val(false);">
        خیر
    </button>
    @if (ViewData.ModelMetadata.IsNullableValueType)
    {
        <button type="button" class="btn btn-info @isIndeterminate bool-@htmlField"
        onclick="$('#@htmlField').val('');">
            نامشخص
        </button>
    }
</div>
```

سورس کامل فایل BootstrapBoolean.cshtml را که در مسیر Views\Shared\EditorTemplates باید کپی شود، در اینجا ملاحظه می‌کنید.

نوع اطلاعاتی که این قالب ادیتور پردازش خواهد کرد از نوع nullable bool است. البته مشکلی هم با نوع‌های bool معمولی ندارد. در حالت nullable، دکمه سوم را به نام «نامشخص» به مجموعه دکمه‌های «بلی» و «خیر» اضافه می‌کند. گاهی از اوقات در فرم‌های دریافت اطلاعات نیاز است بررسی کنیم آیا واقعا کاربر اطلاعاتی را انتخاب کرده یا اینکه بدون توجه به فیلدها، بر روی دکمه ارسال کلیک کرده است. در یک چنین حالتی تعریف دکمه‌های سه وضعیتی Boolean می‌تواند مفهوم پیدا کند. در مورد اصول تهیه این قالب در ابتدای مطلب، با کلاس‌های btn-group و ویژگی data-toggle آشنا شدید. دقیقا این سه دکمه نیز

در اینجا به همین نحو تعریف شده‌اند.

در ابتدای نمایش یک View، خصوصا در حالت ویرایش اطلاعات، نیاز است اطلاعات موجود، به دکمه‌های تعریف شده اعمال شوند. در اینجا برای انتخاب یک دکمه، باید کلاس active به آن نسبت داده شود، که نحوه تدارک آن را در سه متغیر noIsSelected، yesIsSelected و isIndeterminate ابتدای تعاریف قالب مشاهده می‌کنید. سپس یک فیلد مخفی به صفحه اضافه شده است. از این جهت که به کمک jQuery، در حین کلیک بر روی یکی از دکمه‌ها، مقدار آن را به این فیلد که نهایتا به سرور ارسال خواهد شد، اعمال خواهیم کرد.

#### تهیه قالب نمایشی Views\Shared\DisplayTemplates\BootstrapBoolean.cshtml

```
@model bool?
@if (Model.HasValue)
{
    if (Model.Value)
    {
        <span class="label label-success">بلی</span>
    }
    else
    {
        <span class="label label-important">خیر</span>
    }
}
else
{
    <span class="label label-inverse">نامشخص</span>
}
```

در حالت صرفا نمایشی، فایل قالب BootstrapBoolean.cshtml قرار گرفته در مسیر Views\Shared\DisplayTemplates، یک چنین تعاریفی را خواهد داشت.

و نهایتا برای استفاده از آن تنها کافی است توسط ویژگی UIHint، نام این قالب، به خاصیت Boolean مدنظر اعمال شود:

```
[UIHint("BootstrapBoolean")]
public bool? IsActive { set; get; }
```

## نظرات خوانندگان

نویسنده: مهدی پایروند  
تاریخ: ۱۵:۲ ۱۳۹۲/۰۴/۰۴

با سلام، اگر ممکنه sample رو هم بذارین ممنون.

نویسنده: وحید نصیری  
تاریخ: ۱۷:۴۶ ۱۳۹۲/۰۴/۰۴

از اینجا دریافت کنید: [BootstrapBoolean.zip](#)  
+ [نحوه بازسازی وابستگی‌ها](#)

نویسنده: م.ح  
تاریخ: ۱۱:۸ ۱۳۹۲/۰۴/۰۵

با تشکر از آموزش ارائه شده  
پس از اجرای کد فوق مشاهده شد که با کلیک روی گزینه ها، وضعیت active دکمه‌ها تغییر نمی‌کند.  
برای حل مشکل کد زیر را به رویداد onclick اضافه کنید:

```
$(this).siblings().removeClass('active'); $(this).addClass('active');
```

نویسنده: وحید نصیری  
تاریخ: ۱۱:۴۴ ۱۳۹۲/۰۴/۰۵

data-toggle=buttons-radio ذکر شده در مقدمه بحث، زمانی عمل می‌کند (سبب افزوده شدن خودکار کلاس active می‌شود) که فایل‌های اسکریپتی همراه بوت استرپ هم قید شده باشند.  
مراجعه کنید به [کد پیوست شده](#). فایل‌های کامل پروژه موجود هستند. لیست اسکریپت‌ها هم در فایل layout ذکر شده.



عنوان: استفاده از دکمه‌های CSS توئیت‌ر در ASP.NET MVC

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۴/۰۷ ۱۲:۴۴

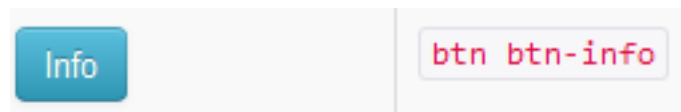
آدرس: [www.dotnettips.info](http://www.dotnettips.info)

برچسب‌ها: ASP.Net, MVC

یک سری نکته ریز را جهت بهبود ظاهر برنامه‌های وب می‌توان در نظر داشت؛ برای مثال:

مجموعه‌ی Twitter Bootstrap که به عنوان یکی از فریم‌ورک‌های خوب CSS مطرح است، دارای تعدادی دکمه تهیه شده با CSS است: ( [^](#) )

برای نمایش یک چنین دکمه‌هایی فقط کافی است یک span را به صفحه اضافه کرده و class آن را مثلا مساوی btn btn-info قرار دهیم تا دکمه‌ای آبی رنگ نمایش داده شود.



طراحی زیبایی دارد. با مرورگرهای جدید سازگار است و ... در اصل یک span بیشتر نیست و قابلیت post back به سرور را ندارد.

برای اضافه کردن چنین قابلیتی می‌توان از نکته زیر استفاده کرد:

```
<span style=" margin:7px;" onclick="$('#loginForm').submit()">ورود به سیستم</span>
```

jQuery که جزء پیش فرض برنامه‌های ASP.NET MVC است. متد submit را هم می‌توان در اینجا در صورت کلیک بر روی یک span فراخوانی کرد. نکته مهم آن ذکر Id فرم جاری است که مثلا به این شکل قابل تعریف است:

```
@using (Html.BeginForm(actionName: "LogOn", controllerName: "Login",
    method: FormMethod.Post, htmlAttributes: new { id = "loginForm" })){
{
}
```

## نظرات خوانندگان

نویسنده: احمد

تاریخ: ۱۱:۱۶ ۱۳۹۱/۰۴/۰۸

بسیار عالی!

وقتی css رو به صفحه اضافه میکنیم روی تمام کنترلها اثر میذاره. که مطلوب نیست. راه حل چیه؟

نویسنده: وحید نصیری

تاریخ: ۱۱:۳۰ ۱۳۹۱/۰۴/۰۸

ضرورتی به استفاده از تمام پکیج آن نیست. مثلاً می‌شود فقط از فایل css مربوط به دکمه‌های آن استفاده کرد تا بر روی سایر کنترل‌ها تاثیر گذار نباشد. مانند این فایل:

[buttons.css](#)

در قسمت اول مشاهده کردیم که به چه صورت می‌توان از دکمه‌های CSS ایی بجای دکمه‌های معمولی در ASP.NET MVC استفاده کنیم. در این قسمت قصد داریم قابلیت مهم زیر را نیز به این تغییر اضافه نمائیم:

کاربر امکان دوبار کلیک کردن همزمان را بر روی دکمه ارسال، نداشته باشد

انجام اینکار با jQuery بسیار ساده است:

```
$(el).prop("onclick", null).attr("onclick", null);
```

به این ترتیب کاربر امکان کلیک مجدد را بر روی دکمه ارسال، از دست خواهد داد. و اینکار ... یک مشکل را اضافه می‌کند:

اگر نتیجه اعتبار سنجی سمت کاربر فرم، تکمیل نشده باشد، کاربر پس از رفع مشکل دیگر نمی‌تواند فرم را ارسال کند. به همین جهت باید به نحوی اطمینان حاصل کرد که آیا اعتبارسنجی موفقیت آمیز بوده یا خیر؛ سپس دکمه را غیرفعال کنیم. برای پیاده سازی این امر باید jQuery Validator همراه با ASP.NET MVC را به صورت دستی فراخوانی کرده و نتیجه آنرا ارزیابی کنیم:

```
function validateThisForm(formId) {
    var val = $(formId).validate();
    val.form();
    return val.valid();
}

function customSubmit(el, formId) {
    if (!validateThisForm(formId)) return;
    $(el).prop("onclick", null).attr("onclick", null);
    $(formId).submit();
}
```

در متد جاوا اسکریپتی validateThisForm، سیستم jQuery Validator به صورت دستی فراخوانی شده و سپس نتیجه عملیات بازگشت داده می‌شود.

در متد customSubmit، ابتدا وضعیت اعتبار سنجی فرم بررسی شده و سپس بر این اساس تصمیم گرفته خواهد شد که آیا باید دکمه غیرفعال شود و سپس فرم ارسال گردد یا خیر.

یک مثال از استفاده آن:

```
@model MvcApplication27.Models.User
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
@using (Html.BeginForm(Html.BeginForm("Index", "Home", FormMethod.Post, new { id = "MyForm" })))
{
    @Html.ValidationSummary(true)

    <fieldset>
        <legend>User</legend>
        <div class="editor-label">
            @Html.LabelFor(model => model.Name)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.Name)
            @Html.ValidationMessageFor(model => model.Name)
        </div>
        <span style="margin-top:7px;" onclick="customSubmit(this, '#MyForm')" class="btn btn-
info">ارسال</span>
    </fieldset>
```

```
}
```

دریافت کدهای کامل این مثال:

[MvcApplication27.7z](#)

اگر مطلب « [استفاده از Twitter Bootstrap در کارهای روزمره طراحی وب](#) » را مطالعه کرده باشید، قسمتی از آن، به فرم‌ها و همچنین جلب توجه کاربران به فیلدها، برای نمایش خطاهای اعتبارسنجی اختصاص داشت. در مطلب جاری قصد داریم تا این موارد را به یک فرم ASP.NET MVC که به صورت پیش فرض از jQuery Validator برای اعتبارسنجی استفاده می‌کند، اعمال کنیم تا حالت نمایشی پیش فرض این فرم‌ها و همچنین خطاهای اعتبارسنجی آن، با Twitter Bootstrap همخوانی پیدا کند.

## مدل برنامه

```
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;

namespace Mvc4TwitterBootstrapTest.Models
{
    public class User
    {
        [DisplayName("نام")]
        [Required(ErrorMessage="لطفا نام را تکمیل کنید")]
        public string Name { set; get; }

        [DisplayName("نام خانوادگی")]
        [Required(ErrorMessage = "لطفا نام خانوادگی را تکمیل کنید")]
        public string LastName { set; get; }
    }
}
```

در اینجا یک مدل ساده را به همراه دو خاصیت و اعتبارسنجی‌های ساده مرتبط با آن‌ها، مشاهده می‌کنید.

## کنترلر برنامه

```
using System.Web.Mvc;
using Mvc4TwitterBootstrapTest.Models;

namespace Mvc4TwitterBootstrapTest.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(User user)
        {
            if (this.ModelState.IsValid)
            {
                if (user.Name != "Vahid")
                {
                    this.ModelState.AddModelError("", "لطفا مشکلات را برطرف کنید");
                    this.ModelState.AddModelError("Name", "نام فقط باید وحید باشد");
                    return View(user);
                }
                // todo: save ...
                return RedirectToAction("Index");
            }
            return View(user);
        }
    }
}
```

کنترلر برنامه نیز نکته مهمی نداشته و بیشتر برای نمایش خطاهای اعتبارسنجی سفارشی این مثال طراحی شده است.

### طراحی View سازگار با Twitter bootstrap

```
@model Mvc4TwitterBootStrapTest.Models.User
@{
    ViewBag.Title = "تعریف کاربر";
}
@using (Html.BeginForm("Index", "Home", FormMethod.Post, new { @class = "form-horizontal" }))
{
    @Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })

    <fieldset>
        <legend>تعریف کاربر</legend>
        <div class="control-group">
            @Html.LabelFor(x => x.Name, new { @class = "control-label" })
            <div class="controls">
                @Html.TextBoxFor(x => x.Name)
                @Html.ValidationMessageFor(x => x.Name, null, new { @class = "help-inline" })
            </div>
        </div>
        <div class="control-group">
            @Html.LabelFor(x => x.LastName, new { @class = "control-label" })
            <div class="controls">
                @Html.TextBoxFor(x => x.LastName)
                @Html.ValidationMessageFor(x => x.LastName, null, new { @class = "help-inline" })
            </div>
        </div>
        <div class="form-actions">
            <button type="submit" class="btn btn-primary">ارسال</button>
        </div>
    </fieldset>
}
```

در اینجا View متناظر با اکشن متد Index را ملاحظه می‌کنید که نکات ذیل به آن اعمال شده است:

- 1) کلاس form-horizontal به فرم جاری اضافه شده است تا در ادامه بتوانیم برچسب‌ها را در کنار تکست باکس‌ها به صورت افقی نمایش دهیم.
- 2) به ValidationSummary کلاس‌های alert alert-error alert-block انتساب داده شده‌اند تا نمایش خطای کلی یک فرم، متناسب با Twitter bootstrap شود.
- 3) هر خاصیت، با یک div دارای کلاس control-group محصور شده است.
- 4) هر برچسب دارای کلاس control-label است.
- 5) به هر ValidationMessageFor کلاس help-inline انتساب داده شده است.
- 6) کنترل‌های ورودی برنامه در div ای با کلاس controls محصور شده‌اند.
- 7) قسمت دکمه فرم، در div ای با کلاس form-actions قرار گرفته تا یک زمینه خاکستری در اینجا ظاهر شود.
- 8) دکمه فرم، با کلاس btn خاص bootstrap تزئین شده.

## تعریف کاربر

نام
لطفا نام را تکمیل کنید

نام خانوادگی
لطفا نام خانوادگی را تکمیل کنید

ارسال

در این حالت به شکل فوق خواهیم رسید. همانطور که ملاحظه می‌کنید در صورتیکه بر روی دکمه ارسال کلیک شود، همان رنگ‌های متداول jQuery Validator ظاهر می‌شوند و کل ردیف همانند روش‌های متداول Twitter bootstrap دارای رنگ قرمز انتساب یافته توسط کلاس error نخواهد شد.

برای رفع این مشکل باید اندکی اسکریپت نویسی کرد:

```
@section javascript
{
    <script type="text/javascript">
        $.validator.setDefaults({
            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');
            }
        });
        $(function () {
            $('form').each(function () {
                $(this).find('div.control-group').each(function () {
                    if ($(this).find('span.field-validation-error').length > 0) {
                        $(this).addClass('error');
                    }
                });
            });
        });
    </script>
}
```

کاری که در اینجا انجام شده، [تغییر پیش فرض‌های](#) jQuery Validator جهت سازگار سازی آن با کلاس error مرتبط با bootstrap است. همچنین در حالت postback و نمایش خطاهای سفارشی، قسمت بررسی field-validation-error انجام شده

و در صورت یافتن موردی، سطر مرتبط با آن، با کلاس error مزین می‌شود.

اینبار در حالت اعتبار سنجی، به شکل ذیل خواهیم رسید:

تعریف کاربر

نام

لطفا نام را تکمیل کنید

نام خانوادگی

لطفا نام خانوادگی را تکمیل کنید

ارسال

و یا اگر کاربر فیلد را تکمیل کند، رنگ فیلد و ردیف تعیین اعتبار شده، سبز می‌شود:

تعریف کاربر

نام

بررسی

نام خانوادگی

لطفا نام خانوادگی را تکمیل کنید

ارسال

و در حالت خطاهای سفارشی سمت سرور، پس از postback، شکل زیر نمایش داده می‌شود:



• لطفا مشکلات را برطرف کنید!

#### تعریف کاربر

نام فقط باید وحید باشد!

بررسی

نام

2

نام خانوادگی

ارسال

## نظرات خوانندگان

نویسنده: وحید نصیری  
تاریخ: ۱۵:۵ ۱۳۹۲/۰۴/۰۵

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

+ [نحوه بازسازی وابستگی‌ها](#)

نویسنده: حمید چگینی  
تاریخ: ۱۴:۱۹ ۱۳۹۲/۰۹/۲۵

ممکنه در مورد `validator.setDefaults.$` بیشتر توضیح بدین. طبق مستندات تنظیمات پیش فرض اصلاح میشه. اما متوجه این مطلب نمی‌شم. دو تابع `highlight` و `unhighlight` از قبل تو این پلاگین وجود دارن و ما اونارو تعریف میکنیم؟ یا اینکه نه ما هرتابعی بخوایم می‌تونیم تعریف کنیم.

نویسنده: وحید نصیری  
تاریخ: ۱۴:۳۳ ۱۳۹۲/۰۹/۲۵

بله. وجود دارند. لیست کامل آن‌ها را در [مستندات آن](#) مطالعه کنید.

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.

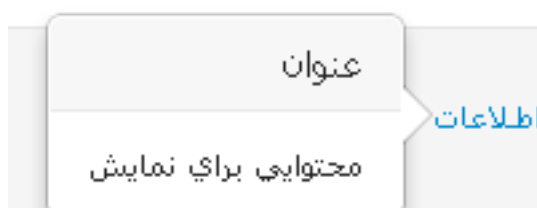
## اصول نمایش Popover در Twitter bootstrap

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

1

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: ۱۴:۴۱ ۱۳۹۲/۰۴/۰۵

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

+ [نحوه بازسازی وابستگی‌ها](#)

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 Tooltip در Bootstrap

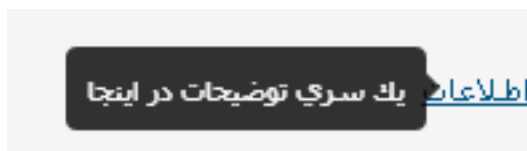
Tooltip یکی دیگر از کامپوننت‌های جاوا اسکریپتی Bootstrap است.

```
<a rel="tooltip" title="یک سری توضیحات در اینجا" href="#">اطلاعات</a>
<script type="text/javascript">
  $(document).ready(function () {
    $('[rel="tooltip"]').tooltip({placement:'top', trigger : 'hover'});
  });
</script>
```

نحوه استفاده از آن را در مثال فوق مشاهده می‌کنید.

برای المان مورد نظر، یک title تعریف کرده‌ایم. برای اینکه tooltip پیش فرض مرورگر در این حالت ظاهر نشود، یک rel=tooltip را در اینجا به المان اضافه کرده و سپس افزونه tooltip بوت استرپ، بر روی کلیه المان‌هایی با rel=tooltip فراخوانی شده است.

placement، محل قرارگیری tooltip را مشخص می‌کند؛ مانند بالا، پایین، چپ و راست. trigger مشخص می‌کند که این tooltip در چه زمانی ظاهر شود. برای مثال در حالت hover یا در حالت focus یک المان. در اینجا بجای title، از ویژگی data-original-title نیز می‌توان استفاده کرد.



## تبدیل خطاهای اعتبارسنجی ASP.NET MVC به Tooltip

هدف ما در اینجا، نهایتاً رسیدن به شکل زیر می‌باشد:



## تعریف کاربر جدید

لطفا نام را تکمیل کنید

نام

نام خانوادگی

ارسال

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

کدهای کامل View برنامه

```
@model Mvc4TwitterBootstrapTest.Models.User
@{
    ViewBag.Title = "تعریف کاربر جدید";
}
@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 تول تپ ندارد
        $(". " + this.settings.validClass).tooltip("destroy");
        // افزودن تول تپ ها
        for (var i = 0; i < errorList.length; i++) {
            var error = errorList[i];
            $(error.element).tooltip({ trigger: "focus" }) // فقط در حالت فوکوس نمایش داده
            شود
            .attr("data-original-title", 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 می کند
$(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 کلاس های ویژه اعتبارسنجی Twitter bootstrap به فرم های ASP.NET MVC](#) » می باشند و از تکرار مجدد آن ها در اینجا صرف نظر گردید.

### توضیحات

- با توجه به اینکه دیگر نمی خواهیم خطاها به صورت برجسته در مقابل کنترلرها نمایش داده شوند، کلیه `Html.ValidationMessageFor` به صورت کامنت درآورده شده اند.
- تغییر دوم مطلب جاری، اضافه شدن متد `showErrors` به تنظیمات پیش فرض `jQuery Validator` است. در این متد، اگر المانی معتبر بود، `Tooltip` آن حذف می شود؛ یا در سایر حالات، المانی که نیاز به اعتبارسنجی سمت کلاینت دارند، یافت شده و سپس ویژگی `data-original-title` با مقداری معادل خطای اعتبارسنجی متناظر، به این المان افزوده شده و سپس متد `tooltip` بوت استرپ بر روی آن فراخوانی می گردد.
- به عبارتی زمانی که یک `input box` در `ASP.NET MVC` به همراه مقادیر مرتبط با اعتبارسنجی آن رندر می شود، 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-original-title` نیز به آن افزوده می گردد:

```

<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="لطفا نام را تکمیل کنید" title="">

```

این مقدار توسط افزونه tooltip بوت استرپ شناسایی شده و مورد استفاده قرار می‌گیرد.

## نظرات خوانندگان

نویسنده: وحید نصیری  
تاریخ: ۱۴:۴۷ ۱۳۹۲/۰۴/۰۵

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

+ [نحوه بازسازی وابستگی‌ها](#)

نویسنده: مهدی پایروند  
تاریخ: ۹:۱۸ ۱۳۹۳/۰۲/۱۰

روش متداول افزودن این tooltip به یک المان جدید که به صفحه اضافه شده چیست؟

نویسنده: وحید نصیری  
تاریخ: ۹:۲۹ ۱۳۹۳/۰۲/۱۰

```
$('#body').tooltip({  
  selector: '[rel=tooltip]'  
});
```

نویسنده: مهدی پایروند  
تاریخ: ۹:۵۴ ۱۳۹۳/۰۲/۱۰

ممنون حل شد. مثال بعد از افزودن لینک زیر:

```
<a id="lnkNo1" href="#" rel="tooltip" title="some text here">some text here too!</a>
```

میتوان با کد زیر tooltip را افزود:

```
$('#lnkNo1').tooltip({ selector: '[rel=tooltip]' });
```

نویسنده: فخاری  
تاریخ: ۱۷:۴۵ ۱۳۹۳/۰۹/۱۲

با سلام

چرا بعد از عملیات ajax ایی دیگه کار نمیده؟

از کد زیر برای نمایش استفاده شده:

```
<script type="text/javascript">  
  $(document).ready(function () {  
    $('[rel="tooltip"]').tooltip({ placement: 'top', trigger: 'hover' });  
  });  
</script>
```

ولی بعد از انجام این دستور دیگه کار نمیده:

```
@Ajax.ActionLink(" ", MVC.Admin.ContactUs.ActionNames.List, MVC.Admin.ContactUs.Name,  
  new  
  {  
    bywriter = ViewBag.bywriter,  
    bydate = ViewBag.bydate,  
    byisread = ViewBag.byisread,  
    byisshow = ViewBag.byisshow,  
    page = max,  
    count = ViewBag.COUNT  
  },  
  new AjaxOptions
```

```
{
    HttpMethod = "Post",
    InsertionMode = InsertionMode.Replace,
    OnBegin = "showLoading",
    UpdateTargetId = "listdiv",
    OnComplete = "hideLoading"
},
new { @class = "glyphicon glyphicon-backward nodecoration", @rel = "tooltip", @title =
"صفحه آخر" })
```

نویسنده:

وحید نصیری

تاریخ:

۱۷:۵۷ ۱۳۹۳/۰۹/۱۲

- لازم هست با مفهومی به نام jQuery Live آشنا باشید: [^](#) و [^](#)
- همچنین در نظرات مطلب جاری در مورد « [المان‌های جدید](#) » یا همان « [المان‌های پویا](#) » بحث شده.

با استفاده از چهارچوب بوت استرپ می‌توان رابط‌های کاربر استاندارد ساخت که قبلاً دوستان در این سایت مطالبی را در این باب نوشته‌اند.

در مطلب [صفحات مودال در بوت استرپ 3](#) در مورد ترکیب قالب بوت استرپ با سیستم اعتبارسنجی MVC و jQuery validation و نمایش فرم‌های مودال بوت استرپ صحبت شده و بسیار کامل هست.

### مشکل:

هنگام کار با بوت استرپ اگر از tab‌های آن در فرم برنامه استفاده کنید، هنگام validate کردن فرم متوجه می‌شوید که فقط کنترل‌های تب جاری اعتبارسنجی میشوند و بدون توجه به سایر کنترل‌هایی که در تب‌های دیگر هستند و احیاناً در وضعیت عدم اعتبار هستند، فرم اعتبارسنجی و کامل اعلام شده و اطلاعات ارسال می‌شود.

### دلیل:

دلیل این اتفاق این هست که jQuery validation به طور پیش فرض کنترل‌هایی که در صفحه مخفی هستند را اعتبارسنجی نمی‌کند و نادیده می‌گیرد.

### راه حل:

با تغییر رفتار پیش فرض سیستم اعتبارسنجی می‌شود این مساله را حل کرد. با اضافه کردن **ignore: ""** اعلام می‌شود که کنترل‌های مخفی هم اعتبارسنجی شوند.

در نهایت کد کامل ترکیب قالب بوت استرپ با سیستم اعتبارسنجی جی کوئری به صورت زیر می‌شود. (فقط همان **ignore: ""** اضافه شده است).

```
function enableBootstrapStyleValidation() {
    $.validator.setDefaults({
        ignore: "",
        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('.form-group').removeClass('has-success').addClass('has-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('.form-group').removeClass('has-error').addClass('has-success');
            }
            $(element).trigger('unhighlighted');
        }
    });
}
```

البته می‌توان این تغییر رفتار پیش فرض را فقط به فرم خاصی هم اعمال کرد. به این صورت

```
$("#form").validate({
    ...
    rules: {...},
    messages: {...},
    ignore: "",
    ...
});
```

## نظرات خوانندگان

نویسنده:

م راد

تاریخ:

۱۳:۳۸ ۱۳۹۲/۱۰/۰۷

با تشکر از ارسال مطلب مفیدتون،

اما موضوعی که پیش میاد اینکه چطوری به ازای هر تب ابتدا اعتبارسنجی فیلدهای اون تب انجام شه و بعد تب بعدی نمایش داده شه؟

نویسنده:

محسن خان

تاریخ:

۱۹:۳۳ ۱۳۹۲/۱۰/۰۷

خوب این همون حالت پیش فرضی هست که توضیح دادن: «هنگام validate کردن فرم متوجه می‌شوید که فقط کنترل‌های تب جاری اعتبارسنجی میشوند و بدون توجه به سایر کنترل‌هایی که در تب‌های دیگر هستند». این tab جاری منظور همون به ازای هر tab جداگانه هست.

برای نمایش خودکار تب بعدی هم باید کدنویسی کنید. فقط کافی هست که کلاس active رو به div تب مورد نظر اضافه کنید:

```
$("#home").removeClass("active"); // this deactivates the home tab  
$("#profile").addClass("active"); // this activates the profile tab
```

همانطور که در مطلب « [اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های ASP.NET MVC](#) » ملاحظه کردید، برای سازگار سازی یک فرم جدید ایجاد شده ASP.NET MVC با پیش فرض‌های Twitter bootstrap، حداقل 8 مرحله باید طی شود و ... چقدر خوب می‌شد اگر این کارها به صورت خودکار توسط VS.NET بجای قالب پیش فرض ایجاد فرم آن، تولید می‌شد. در ادامه قصد داریم این سفارشی سازی را انجام دهیم.

### مراحل کلی سفارشی سازی قالب‌های Scaffolding پیش فرض ASP.NET MVC

قالب‌های Scaffolding پیش فرض ASP.NET در مسیر Microsoft Visual Studio قرار دارند. برای نمونه اگر بخواهیم پیش فرض‌های تولید فرم‌های MVC4 را تغییر دهیم، باید به پوشه MVC 4\CodeTemplates\AddView\CSHTML مراجعه و فایل Create.tt را ویرایش کنیم. اینکار هرچند عملی است اما آنچنان جالب نیست؛ از این جهت که تاثیری کلی و سراسری خواهد داشت. برای اعمال محلی این تغییرات فقط به یک پروژه خاص، تنها کافی است همین مسیر CodeTemplates\AddView\CSHTML به همراه تمام فایل‌های tt آن، در پوشه جاری پروژه مدنظر ما کپی شود. به این ترتیب ابتدا به این پوشه محلی مراجعه خواهد شد. روش دوم کپی کردن این فایل‌ها، استفاده از [بسته نیوگت ذیل](#) است:

```
PM> Install-Package Mvc4CodeTemplatesCSharp
```

### سفارشی سازی فایل Create.tt پیش فرض ASP.NET MVC جهت سازگار سازی آن با Twitter bootstrap

در اینجا قصد داریم همان 8 مرحله مطلب « [اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های ASP.NET MVC](#) » را به فایل Create.tt که اکنون در پوشه CodeTemplates\AddView\CSHTML\Create.tt ریشه پروژه جاری قرار دارد، اعمال کنیم. الف) ابتدا نام این فایل را به CreateBootstrapForm.tt تغییر می‌دهیم. از این لحاظ که این نام جدید در drop down مرتبط با scaffold template صفحه Add view ظاهر خواهد شد. به علاوه نیازی نیست تا این فایل tt در همان لحظه اجرا شود، بنابراین به خواص آن در VS.NET مراجعه کرده و مقدار گزینه custom tool آن را خالی می‌کنیم (مانند سایر فایل‌های tt اضافه شده). ب) قسمت ابتدایی فایل CreateBootstrapForm.tt را که همان کپی مطابق اصل فایل Create.tt است، به نحو ذیل تغییر می‌دهیم:

```
<#
    if (!mvcHost.IsContentPage) {
#>
<script src="~/Scripts/jquery-1.9.1.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

<#
    }
#>
@using (Html.BeginForm()) {
    @Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })

    <fieldset class="form-horizontal">
        <legend><#= mvcHost.ViewDataType.Name #></legend>

<#
foreach (ModelProperty property in GetModelProperties(mvcHost.ViewDataType)) {
    if (!property.IsPrimaryKey && !property.IsReadOnly && property.Scaffold) {
#>
        <div class="control-group">
<#
```



```

        if (property.IsForeignKey) {
#>            @Html.LabelFor(model => model.<#=# property.Name #>, "<#=# property.AssociationName #>", new
{ @class="control-label" })
<#
        } else {
#>            @Html.LabelFor(model => model.<#=# property.Name #>, new { @class="control-label" })
<#
        }
#>

        <div class="controls">
<#
        if (property.IsForeignKey) {
#>            @Html.DropDownList("<#=# property.Name #>", String.Empty)
<#
        } else {
#>            @Html.EditorFor(model => model.<#=# property.Name #>)
<#
        }
#>
        @Html.ValidationMessageFor(model => model.<#=# property.Name #>, null, new { @class="help-
inline" })
</div>
        </div>

<#
    }
}
#>
<div class="form-actions">
    <button type="submit" class="btn btn-primary">ارسال</button>
    <button class="btn">لغو</button>
</div>
</fieldset>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>
<#
if (mvcHost.IsContentPage && mvcHost.ReferenceScriptLibraries) {
#>
@section JavaScript {
}

```

که حاصل آن به صورت ذیل قابل استفاده و دسترسی خواهد بود:

**Add View**

View name:  
Test1

View engine:  
Razor (CSHTML)

☒ Create a strongly-typed view

Model class:  
User (Mvc4TwitterBootStrapTest.Models)

Scaffold template:  
Empty

☐ Create  
**CreateBootstrapForm**  
Delete  
☒ Details  
Edit  
Empty  
List

☒ Reference script libraries

ContentPlaceHolder ID:  
MainContent

Add Cancel

دریافت فایل CreateBootstrapForm.tt اصلاح شده:

[CodeTemplates.zip](#)

همانطور که عنوان شد، برای استفاده از آن فقط کافی است آن را در مسیر  
CodeTemplates\AddView\CSHTML\CreateBootstrapForm.tt ریشه پروژه جاری خود کپی کنید.

## نظرات خوانندگان

نویسنده: محمد آزاد  
تاریخ: ۱۳:۷ ۱۳۹۲/۰۳/۲۹

برای ایجاد یک قالب دلخواه، من به یک مشکل خوردم. فرض کنید من مدلی به این شکل دارم

```
public class MyModel
{
    public Person person { get; set; }
    public string Type{ get; set; }
}
public class Person
{
    Public string FirstName{get;set;}
    public string LastName{get;set;}
}
```

کدی که برای Create توسط قالب ویژه من ایجاد میشه به این صورت هستش

```
@model myModel
@{
    ViewBag.Title = "View3";
}
<h2>View3</h2>

@using (Ajax.BeginForm() {
<section class="Simple Page">
    <div class="row-fluid">
        @Html.LabelFor(model => model.Type, new { @class = "span3" })
        <div class="input-control text span4">
            @Html.EditorFor(model => model.Type, new { @class = "span4 ", placeholder =
            Html.Encode("ResorceName") })
        </div>
        </div>
    </section>
}
```

نویسنده: وحید نصیری  
تاریخ: ۱۴:۸ ۱۳۹۲/۰۳/۲۹

قالب پیش فرض Create از nested properties یا complex properties پشتیبانی نمی‌کند. (البته نه به این معنا که اگر دستی اضافه شوند کار نمی‌کند)

[+ یک راه حل](#) برای این نوع اشیاء توسط یکی از اعضای سابق تیم ASP.NET MVC  
و یا اگر نیاز به اطلاعات بیشتری داشتید، پروژه [MvcScaffolding](#) یک چنین روابطی رو پشتیبانی می‌کند: [MvcScaffolding: One-to-Many Relationships](#)

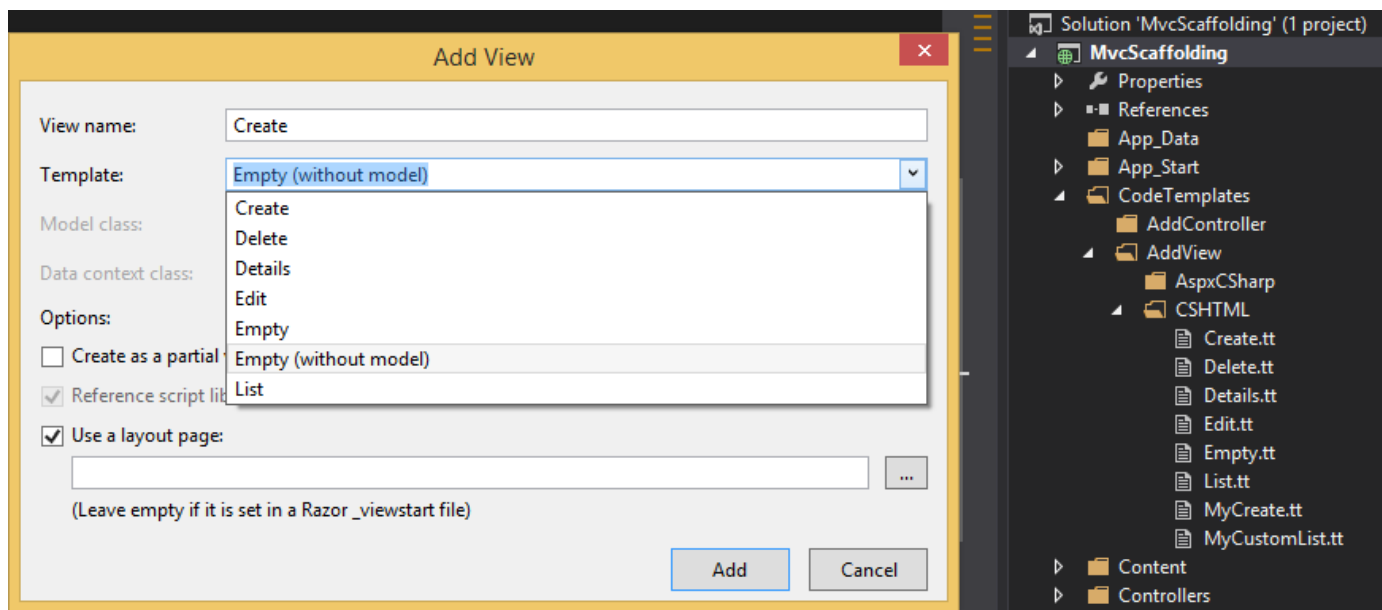
نویسنده: محمد صاحب  
تاریخ: ۱۶:۵۶ ۱۳۹۲/۱۲/۱۰

Compiling transformation: The type or namespace name 'MvcTextTemplateHost' could not be found (are you missing a using directive or an assembly reference?)

دوستان اگه این ارور رو گرفتید با ریستارت کردن ویژوال استادیو ارور رفع میشه

نویسنده: محمد آزاد  
تاریخ: ۱۶:۲ ۱۳۹۳/۰۴/۱۴

سلام. من از vs2013 update 2 استفاده می‌کنم. اما مشکلی که دارم اینه تمپلیت جدید به لیست scaffold template اضافه نمیشه؟ خواستم بدونم تو این ورژن باید کار خاصی انجام بدم؟  
بعد از ایجاد تمپلیت جدید در دیالوگ AddView اسم تمپلیتم رو نمیبینم. ساختار فولدر بندی در ریشه پروژم وجود داره  
CodeTemplates\AddView\CSHTML\MyCreate.tt



نویسنده: محمد آزاد  
تاریخ: ۱۷:۱۳ ۱۳۹۳/۰۴/۱۴

جواب رو پیدا کردم توی 5 mvc ساختار فولدرها متفاوت هستش <http://weblogs.asp.net/imranbaloch/customizing-the-asp-net-mvc-5-web-api-2-scaffolding-templates>

پیش نیاز مطلب جاری مطالب زیر می‌باشند:

1- [EF Code First #8](#)

2- [مباحث تکمیلی مدل‌های خود ارجاع دهنده در EF Code First](#)

3- [نگاهی به اجزای تعاملی Twitter Bootstrap](#)

هدف از مطلب جاری نحوه نمایش منوهای چند سطحی می‌باشد، ابتدا مثال کامل زیر را در نظر بگیرید :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Menu.Models.Entities
{
    public class Category
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public int? ParentId { get; set; }
        public virtual Category Parent { get; set; }
        public virtual ICollection<Category> Children { get; set; }
    }
}

public class MyContext : DbContext
{
    public DbSet<Category> Category { get; set; }

    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        // Self Referencing Entity
        modelBuilder.Entity<Category>()
            .HasOptional(x => x.Parent)
            .WithMany(x => x.Children)
            .HasForeignKey(x => x.ParentId)
            .WillCascadeOnDelete(false);

        base.OnModelCreating(modelBuilder);
    }
}
```

همانطور که ملاحظه می‌کنید، مدل ما شامل مشخصات گروه محصولات می‌باشد که به صورت خود ارجاع دهنده (خاصیت Parent به همین کلاس اشاره میکند) تعریف شده است. در مورد خواص مدل‌های خود ارجاع دهنده، مطالبی را در سایت مطالعه کردید (خواص مربوط در مطالب گفته شده دقیقاً به همان صورت می‌باشد و نیازی به توضیح اضافه‌تری نیست).

هدف از این بحث، نحوه نمایش گروه محصولات داخل منو به صورت چند سطحی می‌باشد، جهت نمایش می‌بایست از تکنیک recursive function استفاده کنید، ابتدا در نظر داشته باشید که ساختار منوی تشکیل شده می‌بایست بدین صورت باشد :



این حالت می‌تواند تا n سطح پیش برود، حال نحوه نمایش در View مربوطه باید به صورت زیر باشد :

```
@using Menu.Helper
@model IEnumerable<Models.Entities.Category>
@ShowTree(Model)

@helper ShowTree(IEnumerable<Menu.Models.Entities.Category> categories)
{
    foreach (var item in categories)
    {
        <li class="@((item.Children.Any()) ? "dropdown-submenu" : "")">
            @Html.ActionLink(item.Name, actionName: "Category", controllerName: "Product", routeValues: new
            { Id = item.Id, productName = item.Name.ToSeoUrl() }, htmlAttributes: null)
            @if (item.Children.Any())
            {
                <ul>
                    @ShowTree(item.Children)
                </ul>
            }
        </li>
    }
}
```

توجه داشته باشید که رندر نهایی توسط Bootstrap انجام شده است. ساختار منو همانطور که ملاحظه می‌کنید با استفاده از کلاس‌های drop-down که از کلاس‌های پیش فرض بوت استرپ می‌باشد تشکیل شده است همچنین کلاس dropdown-submenu که از نسخه 2 به بعد بوت استرپ موجود می‌باشد، استفاده شده است.

#### یک نکته :

در خط 9 این مورد را که آیا آیتم جاری فرزندی دارد چک کرده ایم اگر داشته باشد کلاس dropdown-submenu را به li جاری اضافه میکند.

## نظرات خوانندگان

نویسنده: سرزمین خورشید  
تاریخ: ۱۱:۱۱ ۱۳۹۲/۰۷/۲۲

با سلام

من این برنامه رو اجرا کردم ولی روی خط 5 پیغام خطا System.NullReferenceException: Object reference not set to an instance of an object

رو میگیرم .

اگر امکانش هست میتونید پروژه این مثال رو بگذارید.

ممنون

نویسنده: سیروان عقیقی  
تاریخ: ۱۱:۵۶ ۱۳۹۲/۰۷/۲۲

شما Query رو به View مربوطه پاس دادید؟

من توی لایه Service متد GetAll رو به این صورت تعریف کردم :

```
public IList<Category> GetAll()
{
    //return _category.AsNoTracking().ToList();
    return _category.Where(category => category.ParentId == null)
        .Include(category => category.Children).ToList();
}
```

و در نهایت هم توسط Action Method زیر اون رو به Partial View پاس دادم :

```
[ChildActionOnly]
public ActionResult Categories()
{
    var query = _categoryService.GetAll();
    return PartialView("_Categories", query);
}
```

نویسنده: مهران کلانتری  
تاریخ: ۱۲:۱۱ ۱۳۹۲/۰۷/۳۰

سلام

اگر امکان دارد مثالی از درج ، ویرایش یا حذف زیر مجموعه‌ها را بیان کنید یا اگر منبع مفیدی می‌شناسید برای ویرایش و کار با navigation peroperty لطفا معرفی کنید

با سپاس فراوان

نویسنده: سیروان عقیقی  
تاریخ: ۱۴:۲ ۱۳۹۲/۰۷/۳۰

سلام،

به عنوان مثال من برای ثبت یک Category جدید از [TreeView](#) استفاده میکنم به اضافه یک مورد که تعیین کننده Child یا Parent بودن Category است.

نویسنده: صابر فتح الهی  
تاریخ: ۲۳:۵۴ ۱۳۹۲/۰۹/۰۳

جناب مهندس ممنون از زحمت شما.

یک سوال؟

در بوت استرپ منوی چند سطحی نمی‌توان درست کرد آیا کد خاصی اضافه کردین یا از پلاگین خاصی استفاده میکنین؟

نویسنده: سیروان عقیقی

تاریخ: ۱۳۹۲/۰۹/۰۴

من در این مثال از ورژن 2 بوت استرپ استفاده کرده ام که دارای کلاس dropdown-submenu می‌باشد. این کلاس ظاهراً در ورژن 3 حذف شده است، در هر صورت اگر از ورژن 3 استفاده می‌کنید می‌توانید با افزودن مقداری کد CSS این مورد را اضافه کنید:

```
.dropdown-submenu{position:relative;}
.dropdown-submenu>.dropdown-menu{top:0;left:100%;margin-top:-6px;margin-left:-1px;-webkit-border-radius:0 6px 6px 6px;-moz-border-radius:0 6px 6px 6px;border-radius:0 6px 6px 6px;}
.dropdown-submenu:hover>.dropdown-menu{display:block;}
.dropdown-submenu>a:after{display:block;content:" ";float:right;width:0;height:0;border-color:transparent;border-style:solid;border-width:5px 0 5px 5px;border-left-color:#cccccc;margin-top:5px;margin-right:-10px;}
.dropdown-submenu:hover>a:after{border-left-color:#ffffff;}
.dropdown-submenu.pull-left{float:none}.dropdown-submenu.pull-left>.dropdown-menu{left:-100%;margin-left:10px;-webkit-border-radius:6px 0 6px 6px;-moz-border-radius:6px 0 6px 6px;border-radius:6px 0 6px 6px;}
```

نویسنده: رضایی

تاریخ: ۱۳۹۳/۰۳/۰۲ ۱۴:۲۵

با سلام و تشکر به خاطر مطلب مفیدتون

زمانی که منو رو به صورت پویا ایجاد می‌کنیم در هر بار لود صفحه باید اطلاعات مربوط به منو از بانک اطلاعاتی دریافت بشه آیا اطلاعات کش میشه از روش خاصی استفاده میشه که در هر بار لود صفحه درخواست به بانک ارسال نشه

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۳/۰۲ ۱۴:۴۰

این مسایل را خودتان باید مدیریت کنید. یا از سطح دوم کش استفاده کنید یا از روش‌های متداول کش کردن اطلاعات در دات نت.

برای مطالعه بیشتر:

- [MVC #19](#)

- [چگونه نرم افزارهای تحت وب سریعتری داشته باشیم؟ قسمت دوم](#)

- [Implementing second level caching in EF code first](#)

- [استفاده از AOP Interceptors برای حذف کدهای تکراری کش کردن اطلاعات در لایه سرویس برنامه](#)

و ...

نویسنده: رضایی

تاریخ: ۱۳۹۳/۰۳/۰۳ ۰۵:۵۴

ممنون! از حالت AOP Interceptors استفاده کردم خیلی جالب بود.

فقط یه راهنمایی می‌خواستم به نظر شما برای کش کردن منوهای پویا چه زمانی میتونه مناسب باشه؟

[CacheMethod(SecondsToCache = ?)]

باز هم تشکر



نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۳/۰۳ ۱:۲

در سایت جاری اطلاعات آماری پویای منوی سمت راست صفحه، 15 دقیقه کش می‌شوند.

نویسنده: حیدر ماموسیان  
تاریخ: ۱۳۹۳/۰۳/۰۶ ۱۲:۵۶

با سلام  
ضمن تشکر از سایت بسیار خوبتون  
اگر ممکن است در خصوص متد ToSeoUrl  
که در خط

```
@Html.ActionLink(item.Name, actionName: "Category" , controllerName: "Product" , routeValues: new { Id  
= item.Id, productName = item.Name.ToSeoUrl() }, htmlAttributes: null )
```

بکار رفته توضیح بدید

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۳/۰۶ ۱۳:۰

همان متد [GenerateSlug](#) است.

نویسنده: مهرداد  
تاریخ: ۱۳۹۳/۰۷/۰۳ ۱۲:۲۹

سلام؛ من مطالبی را که ابتدای مقاله فرمودید، مطالعه کرده‌ام. اگر امکان دارد مثال پروژه را قرار دهید تا بتوانیم از آن استفاده کنیم.

نویسنده: رضایی  
تاریخ: ۱۳۹۳/۰۷/۲۸ ۱۳:۱۵

سلام؛ در این روش (AOP Interceptors) آیا میتوان موقع افزودن خبر جدید یا منو جدید یا ... کش سرویس مورد نظر رو غیرفعال کرد و پس از اولین لود دوباره اطلاعات کش شوند

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۷/۲۸ ۱۳:۳۳

» - [CacheManager.InvalidateCache](#) «

» - [EfHttpContextProvider.InvalidateSecondLevelCache](#) «

DOM در حالت عادی بسیار نامرتب است. همچنین با افزودن کلاس‌های CSS، کد HTML به مراتب نامرتب‌تر از قبل می‌شود. بوت استرپ نیز شامل تعداد زیادی از کلاس‌های CSS می‌باشد که برای انجام وظایف خاصی به HTML اضافه می‌شوند. روش متداول استفاده از بوت استرپ

### Embedd کردن کلاس‌های CSS بوت استرپ به صورت مستقیم درون HTML

اغلب فریم‌ورک‌ها، از لحاظ معنایی یا semantic، دارای مشکل هستند. اگر به سورس HTML صفحاتی که با این نوع از فریم‌ورک‌ها ساخته شده باشند نگاهی بیندازید با حجم زیادی از کلاس‌هایی مانند `<div class="row">` و یا `<div class="col-sm">` مواجه خواهید شد. نوشتن کدهای HTML به این صورت از لحاظ معنایی اشتباه است. مثلاً اگر بنا به دلایلی سازندگان بوت استرپ تصمیم بگیرند نام کلاس‌های را در نسخه بعدی این فریم‌ورک تغییر دهند (مانند تغییر نام کلاس‌ها در نسخه‌ی 3 بوت استرپ)، و یا اگر در آینده بخواهید از یک فریم‌ورک دیگر در سایت‌تان استفاده کنید. باید این تغییرات را در تمام صفحات سایت‌تان اعمال کنید؛ در نتیجه اینکار زمان زیادی را از شما صرف می‌کند.

راه‌حل؟

### استفاده از CSS preprocessors

بوت استرپ، از [Less](#) برای اینکار استفاده می‌کند. Less در واقع یک CSS preprocessor نوشته شده با جاوا اسکریپت است که قابلیت اجرا در مرورگر را دارد. Less امکانات زیادی، از قبیل استفاده از توابع، متغیرها، Mixins و ... را در اختیار شما قرار می‌دهد. در واقع هدف از Less، نگهداری آسان و قابلیت توسعه فایل‌های CSS می‌باشد. در این حالت شما کدهای CSS خود را درون فایل‌هایی با پسوند Less می‌نویسید. در این حالت بجای پیوست کردن کلاس‌های بوت استرپ در کد HTML، آن را درون استایل‌شیت پیوست خواهید کرد. همانطور که عنوان شد، بوت استرپ با Less نوشته شده است. فایل‌های Less بوت استرپ را می‌توانید از مخزن کد [گیت‌هاب](#) آن دانلود نمایید یا اینکه از طریق [نیوگت](#) می‌توانید آن را نصب کنید:

```
PM> Install-Package Twitter.Bootstrap.Less
```

کار با Less خیلی ساده است. به عنوان مثال در کد زیر یک کلاس با نام loud داریم که استایل‌هایی را به آن اعمال کرده‌ایم. اکنون جهت استفاده مجدد از این استایل‌ها برای کلاسی دیگر، نیاز به نوشتن مجدد آن نیست. کافی همانند یک تابع در هر کلاسی آن را فراخوانی کنیم:

```
.loud {
  color: red;
}

// Make all H1 elements loud
h1 {
  .loud;
}
```

**نکته:** در Visual Studio 2012 Update 2 به بعد به صورت توکار از فایل‌های Less [پشتیبانی](#) می‌شود. (توسط پلاگین [Web Essentials](#))

### استفاده از Mixins

با استفاده از Mixins می‌توانیم عناصر داخل صفحات مان را به صورت [Semantic](#) تعریف نمائیم. به عنوان مثال می‌خواهیم با استفاده از سیستم گرید بوت استرپ، ساختاری مانند تصویر زیر را داشته باشیم:

Content Main  
8 columns wide

Content  
Secondary  
4 columns wide

در حالت معمول با استفاده از کلاس‌های CSS بوت استرپ می‌توانیم اینکار را انجام دهیم:

```
<div class="container">
  <div class="row">
    <div class="col-md-8">
      Content - Main
    </div>
    <div class="col-md-4">
      Content - Secondary
    </div>
  </div>
</div>
```

کد فوق را بهتر است به این صورت بنویسیم:

```
<div class="wrapper">
  <div class="content-main">
    Content - Main
  </div>
  <div class="content-secondary">
    Content - Secondary
  </div>
</div>
```

در بوت استرپ از [Less Mixins](#) جهت اعمال استایل‌هایی مانند row و column می‌توانیم استفاده کنیم. به طور مثال برای اعمال استایل به کلاس‌های فوق می‌توانیم به این صورت عمل کنیم:

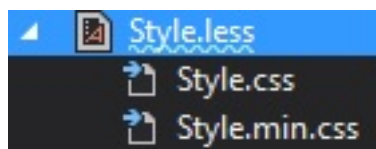
```
// Core variables and mixins
@import "variables.less";
@import "mixins.less";
.wrapper {
  .make-row();
}
.content-main {
```

```
.make-lg-column(8);
}
.content-secondary {
  .make-lg-column(3);
  .make-lg-column-offset(1);
}
```

کد فوق بعد از کامپایل به کد زیر تبدیل خواهد شد:

```
.wrapper {
  margin-left: -15px;
  margin-right: -15px;
}
.content-main {
  position: relative;
  min-height: 1px;
  padding-left: 15px;
  padding-right: 15px;
}
@media (min-width: 1200px) {
  .content-main {
    float: left;
    width: 66.66666666666666%;
  }
}
.content-secondary {
  position: relative;
  min-height: 1px;
  padding-left: 15px;
  padding-right: 15px;
}
@media (min-width: 1200px) {
  .content-secondary {
    float: left;
    width: 25%;
  }
}
@media (min-width: 1200px) {
  .content-secondary {
    margin-left: 8.333333333333332%;
  }
}
```

همانطور که قبلاً عنوان شد ویژوال استودیو به راحتی توسط افزونه Web Essentials از فایل‌های Less پشتیبانی می‌کند. در نتیجه کامپایل فایل‌های Less داخل ویژوال استودیو توسط این افزونه به راحتی قابل انجام می‌باشد. یک قابلیت جالب دیگر در رابطه با فایل‌هایی Less، تولید نسخه‌های CSS عادی و فشرده نهایی توسط افزونه Web Essentials می‌باشد. به طور مثال شما می‌توانید نسخه minified شده را به Layout تان اضافه کنید. بعد از هربار تغییر در فایل Less این فایل نیز به روز خواهد شد:



همچنین برای دیگر اجزای بوت‌استرپ نیز می‌توانید به این صورت عمل کنید:

```
<!-- Before -->
<a href="#" class="btn danger large">Click me!</a>

<!-- After -->
<a href="#" class="annoying">Click me!</a>

a.annoying {
  .btn;
  .btn-danger;
```

```
.btn-large;  
}
```

خب، با استفاده از این حالت، کدهای HTML به صورت مرتب‌تر، قابل انعطاف‌تر و همچنین از لحاظ معنایی (Semantic) استاندارد خواهند بود. بنابراین با آمدن یک فریم‌ورک جدید، به راحتی امکان سوئیچ کردن برای ما میسر و آسان‌تر از قبل خواهد شد.

## نظرات خوانندگان

نویسنده: سعید شیرزادیان

تاریخ: ۱۳۹۳/۰۴/۰۳ ۰:۰

سلام. مطلب جالبی بود. آیا قابلیت فوق برای حالت بدون Less امکان دارد. من در یک پروژه از بوت استرپ استفاده کردم و یکسری تغییرات فارسی نیز در آن اعمال شده است. راهکاری دارید؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۴/۰۳ ۰:۵۷

نسخه‌ی LESS راست به چپ بوت استرپ هم موجود است .  
ضمناً نگارش 3.2 بوت استرپ قرار است که به صورت رسمی RTL پشتیبانی کند.

نویسنده: حسن محمدی

تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۳:۳۵

با سلام و تشکر از سایت بی نهایت خوبتون

مقاله عالی بود اگر امکانش بود به تمپلیت کوچک سایتو بوت استرپ و less بگید بی نهایت ممنون میشم. من خودم تو استفاده از less خیلی مشکل دارم خیلی‌ها هم مثل خودم می‌شناسنم. من آپدیت 2 ویژوال استودیو 2013 و webessenial رو نصب کردم تا فایل‌های less رو برام باز کرد.

فقط bootstrap.less رو بدون مشکل باز میکنم و نسخه css رو میاره ولی مثلاً هر فایل دیگه ای مثل alerts.less رو باز میکنم پیغام خطا میده که مثلاً variable @alert-padding is undefined یا چیزهای دیگرو میگه undefined به نظرتون مشکل از چیه؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۳:۴۵

مشکلی ندارد . نسخه‌ی less بوت استرپ از چندین و چند فایل تشکیل شده که نهایتاً از کامپایل این‌ها یک فایل نهایی تولید می‌شود. برای مثال اگر فایل [bootstrap.less](#) را باز کنید، ترتیب import فایل‌ها قابل ملاحظه هستند. ابتدا فایل [variables.less](#) ذکر شده، چون مقادیر پیش فرض متغیرهای سایر فایل‌ها در آن موجود است و همینطور الی آخر. فایل نهایی که سبب کامپایل تمام تغییرات خواهد شد، فایل [bootstrap.less](#) است.

نویسنده: حسن محمدی

تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۳:۵۸

ممنون که پاسخ دادید من اگر بخوام مثلاً navbar رو تغییر بدم باید bootstrap.less رو ویرایش کنم یا navbar.less ؟  
بعد چیکار کنم که خطا برطرف شه و این ترتیبو بفهمم و مثلاً navbar.less رو باز میکنم دیگه خطا نده؟  
ممنون

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۴:۲۶

- هیچکدام. فایل navbar.less تشکیل شده از یک سری متغیر. این متغیرها در فایل variables.less مقدار دهی شده‌اند. بنابراین برای تغییر آن باید فایل variables.less ویرایش شوند (در این فایل، Navbar را جستجو کنید).  
- یک کپی از فایل اصلی bootstrap.less را مثلاً به نام test.less ایجاد کنید (با همان محتوا). المان‌های مختلف آن را حذف کنید تا به حداقل وابستگی‌هایی که برای کامپایل navbar.less نیاز است برسید:

```
// Core variables and mixins
@import "variables.less";
@import "mixins.less";
```

```
// Core CSS
@import "forms.less";

// Components
@import "navbar.less";

// Utility classes
@import "utilities.less";
```

در بوت استرپ برای نمایش اعلانی به کاربر، از کلاس alert می‌توان استفاده کرد. برای نمایش این اعلان کافی است محتوای خود را درون یک div با کلاس alert قرار دهیم:

```
<div class="alert">
    نمایش اعلانات
</div>
```

نمایش اعلانات

تعدادی کلاس دیگر نیز جهت استفاده از رنگ‌های مختلف نیز توسط بوت استرپ ارائه شده است:

نمایش اعلانات - کلاس alert-block

نمایش اعلانات - کلاس alert-danger

نمایش اعلانات - کلاس alert-error

نمایش اعلانات - کلاس alert-info

نمایش اعلانات - کلاس alert-success

همچنین اگر مایل بودید می‌توانید با افزودن یک دکمه با کلاس close و ویژگی data-dismiss مساوی alert، امکان بستن پیام را در اختیار کاربر قرار دهید:

```
<div class="alert alert-warning alert-dismissible">
    <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
```



```
<div>
نمایش اعلان
</div>
```

در ادامه قصد داریم این پیام را بعد از ثبت اطلاعات، به کاربر نمایش دهیم. یعنی در داخل کد، امکان صدا زدن این نوع پیام‌ها را داشته باشیم.

ابتدا کلاس‌های زیر را تعریف می‌کنیم:

#### کلاس Alert

```
public class Alert
{
    public const string TempDataKey = "TempDataAlerts";
    public string AlertStyle { get; set; }
    public string Message { get; set; }
    public bool Dismissable { get; set; }
}
```

در کلاس فوق خصوصیت یک alert را تعریف کرده‌ایم (از خاصیت TempDataKey جهت پاس دادن alertها به view استفاده می‌کنیم).

#### کلاس AlertStyles

```
public class AlertStyles
{
    public const string Success = "success";
    public const string Information = "info";
    public const string Warning = "warning";
    public const string Danger = "danger";
}
```

کلاس فوق نیز جهت نگهداری اسامی کلاس‌های alert، مورد استفاده قرار می‌گیرد. قدم بعدی، استفاده از کلاس‌های فوق و انتقال alerts توسط TempData به داخل view می‌باشد. برای جلوگیری از زیاد شدن حجم کدهای تکراری داخل کنترلرها و همچنین به عنوان یک Best practice، یک کنترلر Base را برای اینکار تعریف می‌کنیم و متدهای موردنیاز را داخل آن می‌نویسیم:

```
public class BaseController : Controller
{
    public void Success(string message, bool dismissable = false)
    {
        AddAlert(AlertStyles.Success, message, dismissable);
    }

    public void Information(string message, bool dismissable = false)
    {
        AddAlert(AlertStyles.Information, message, dismissable);
    }

    public void Warning(string message, bool dismissable = false)
    {
        AddAlert(AlertStyles.Warning, message, dismissable);
    }

    public void Danger(string message, bool dismissable = false)
    {
        AddAlert(AlertStyles.Danger, message, dismissable);
    }

    private void AddAlert(string alertStyle, string message, bool dismissable)
    {
        var alerts = TempData.ContainsKey(Alert.TempDataKey)
            ? (List<Alert>)TempData[Alert.TempDataKey]
            : new List<Alert>();

        alerts.Add(new Alert
        {
            AlertStyle = alertStyle,
            Message = message,
        });
    }
}
```

```

        Dismissable = dismissable
    });
    TempData[Alert.TempDataKey] = alerts;
}
}

```

از متدهایی که به صورت عمومی تعریف شده‌اند جهت ارسال پیام به view استفاده می‌کنیم. متد AddAlert نیز جهت ایجاد لیستی از پیام‌ها (Alert) مورد استفاده قرار می‌گیرد؛ زیرا ممکن است بخواهید هم زمان از چندین متد عمومی فوق استفاده کنید، یعنی چندین پیام را به کاربر نمایش دهید.

**نکته:** در کد فوق از TempData جهت پاس دادن شیء alerts استفاده کرده‌ایم. TempData به صورت short-lived عمل می‌کند به دو دلیل: 1- بلافاصله بعد از خوانده شدن، حذف خواهد شد. 2- پس از پایان درخواست از بین خواهد رفت. از TempData جهت پاس دادن داده‌ها از درخواست فعلی به درخواست بعدی (redirect از یک صفحه به صفحه دیگر) استفاده می‌شود. یعنی در زمان redirect سعی می‌کند داده‌های بین redirectها را در خود نگه دارد. اگر از ViewBag و ViewData استفاده می‌کردیم داده‌های داخل آنها بلافاصله بعد از redirect شدن null می‌شدند. به طور مثال اکشن متد زیر را در نظر بگیرید:

```

public ActionResult Index()
{
    var userInfo = new
    {
        Name = "Sirwan",
        LastName = "Afifi",
    };

    ViewData["User"] = userInfo;
    ViewBag.User = userInfo;
    TempData["User"] = userInfo;

    return RedirectToAction("About");
}

```

: view

```

@{
    ViewBag.Title = "About";
}

<h1>Tempdata</h1><p>@TempData["User"]</p>
<h1>ViewData</h1><p>@ViewData["User"]</p>
<h1>ViewBag</h1><p>@ViewBag.User</p>

```

اگر کد فوق را تست کنید خواهید دید که در خروجی تنها اطلاعات داخل TempData نمایش داده می‌شود. معمولاً برای ارسال داده‌های خطاها از TempData استفاده می‌شود.

اکنون در هر کنترلری که می‌خواهید پیامی را به صورت alert، پس از ثبت اطلاعات به کاربر نمایش دهید، باید از کنترلر BaseController ارث‌بری کنید:

```

public class NewsController : BaseController
{
    readonly INewsService _newsService;
    readonly IUnitOfWork _uow;
    public NewsController(INewsService newsService, IUnitOfWork uow)
    {
        _newsService = newsService;
        _uow = uow;
    }
    [HttpPost]
    [ValidateAntiForgeryToken]
    [ValidateInput(false)]
    public ActionResult Create(News news)
    {

```

```

        if (ModelState.IsValid)
        {
            _newsService.AddNews(news);
            _uow.SaveChanges();
            Success(string.Format("با موفقیت ذخیره گردید <b>{0}</b> خبر با عنوان", news.Title),
true);
            return RedirectToAction("Index");
        }
        Danger("خطا در هنگام ثبت اطلاعات");
        return View(news);
    }
    [HttpPost]
    public ActionResult Delete(int id)
    {
        _newsService.DeleteNewsById(id);
        _uow.SaveChanges();
        Danger("اطلاعات مورد نظر با موفقیت حذف گردید", true);
        return RedirectToAction("Index");
    }
}

```

## نمایش پیام‌ها

برای نمایش پیام‌ها یک partial view با نام Alerts\_ در مسیر Views\Shared ایجاد می‌کنیم:

```

@{
    var alerts = TempData.ContainsKey(Alert.TempDataKey)
        ? (List<Alert>)TempData[Alert.TempDataKey]
        : new List<Alert>();

    if (alerts.Any())
    {
        <hr />
    }

    foreach (var alert in alerts)
    {
        var dismissableClass = alert.Dismissable ? "alert-dismissable" : null;
        <div class="alert alert-@alert.AlertStyle @dismissableClass">
            @if (alert.Dismissable)
            {
                <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button>
            }
            @Html.Raw(alert.Message)
        </div>
    }
}

```

در کد فوق، ابتدا شیء alerts را از TempData دریافت کرده‌ایم و توسط یک حلقه foreach، داخل آن به ازای هر آیتم، آن را پیمایش می‌کنیم و در نهایت کد html متناظر با هر alert را در خروجی نمایش می‌دهیم. اکنون جهت استفاده از partial view فوق در جایی که می‌خواهید پیام نمایش داده شود partial view فوق را فراخوانی کنید (به عنوان مثال داخل فایل Layout):

```

<div>
    @{ Html.RenderPartial("_Alerts"); }
    @RenderBody()
</div>

```



خبر با عنوان خبر اول با موفقیت ذخیره گردید!



اطلاعات مورد نظر با موفقیت حذف گردید!

## نظرات خوانندگان

نویسنده: محمد رعیت پیشه  
تاریخ: ۲۳:۴۷ ۱۳۹۳/۰۴/۰۵

آیا تاحالا این روش رو در فرم‌های Modal هم امتحان کردید؟ فکر می‌کنم با یک مشکل برخورد خواهید کرد اونم اینکه پیغام در تمام صفحات زیرین هم نمایش داده خواهد شد، همچنین در Partial View ها باید مکانیزمی به کار برده شود که پیغام در موقعیت تعیین شده نمایش داده شود و نه در تمام جاهایی از صفحه که

```
@{ Html.RenderPartial("_Alerts"); }
```

به کار رفته است.

نویسنده: سیروان عقیفی  
تاریخ: ۲۳:۵۴ ۱۳۹۳/۰۴/۰۵

- در اینجا هدف نمایش پیام‌ها در صفحه ایی دیگر است.  
- پیام‌ها زمانی نمایش داده می‌شوند که TempData حاوی اطلاعات ارسالی باشد. فراخوانی partial نیز یکبار درون Layout انجام می‌شود.

نویسنده: مهدی  
تاریخ: ۱۱:۴۴ ۱۳۹۳/۰۴/۰۶

آیا این روش رو میشه برای صفحاتی که بصورت Ajax کار میکنند استفاده کرد؟  
این روش شما تنها در صورت Refresh صفحه کار میکنه

نویسنده: سیروان عقیفی  
تاریخ: ۱۳:۵۷ ۱۳۹۳/۰۴/۰۶

بله؛ می‌توانید به عنوان مثال در view ی create تان از ای جکس استفاده کنید:

```
<div id="result"></div>
@using (Ajax.BeginForm(new AjaxOptions
{
    HttpMethod = "POST",
    InsertionMode = InsertionMode.Replace,
    UpdateTargetId = "result"
}))
```

نویسنده: محمد صاحب  
تاریخ: ۱۷:۱۵ ۱۳۹۳/۰۶/۱۶

برای مواقعی که سشن رو خاموش کردیم بین ریدایرکت‌ها این روش جواب نمیده آیا راه جایگزینی هست؟

نویسنده: سیروان عقیفی  
تاریخ: ۱۷:۲۱ ۱۳۹۳/۰۶/۱۶

می‌توانید رفتار TempData را در حالت خاموش بودن سشن با پیاده‌سازی اینترفیس TempDataProvider تغییر دهید. (+ , +)

نویسنده: محمد صاحب  
تاریخ: ۱۷:۴۶ ۱۳۹۳/۰۶/۱۶

مرسی

با فقط خواندنی کردن سشن هم جواب نداد !

```
[SessionState(SessionStateBehavior.ReadOnly)]
```

میمونه نوشتن پروایدر که باید براش وقت بذارم.

چندی پیش در همین وبسایت مطلبی تحت عنوان «[اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های ASP.NET MVC](#)» منتشر شد. این مقاله مرتبط با نسخه دوم فریم‌ورک محبوب Bootstrap بود. قصد داریم به بازنویسی کدهای مرتبط بپردازیم و کلاس‌های مرتبط با [نسخه سوم این فریم‌ورک](#) را هم با فرم‌های خودمان سازگار کنیم. مثل مقاله‌ی ذکر شده توضیحات را با یک مثال همراه می‌کنم.

#### مدل برنامه

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace FormValidationWithBootstrap.Models
{
    [Table("Product")]
    public class ProductModel
    {
        [Key]
        public int Id { get; set; }
        [Required(ErrorMessage = "{0} باید آن را وارد کنید {0}")]
        [StringLength(50, ErrorMessage = "{1} کاراکتر باشد")]
        [Display(Name = "نام کالا")]
        public string Name { get; set; }
        [Required(ErrorMessage = "{0} باید آن را وارد کنید {0}")]
        [Display(Name = "قیمت")]
        [DataType(DataType.Currency)]
        public double Price { get; set; }
        [Required(ErrorMessage = "{0} باید آن را وارد کنید {0}")]
        [Display(Name = "موجودی")]
        public int Qty { get; set; }
    }
}
```

قرار هست که جدولی داشته باشیم با نام Product برای ثبت محصولات. مدل برنامه شامل خاصیت‌های مرتبط و همچنین اعتبارسنجی‌های مد نظر ما هست.

#### کنترلر برنامه

```
using System.Web.Mvc;
using FormValidationWithBootstrap.Models;

namespace FormValidationWithBootstrap.Controllers
{
    public class ProductController : Controller
    {
        // GET: Product
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult New()
        {
            return View();
        }

        [HttpPost]
        public ActionResult New(ProductModel product)
        {
            if (!ModelState.IsValid)
                return View(product);

            if (product.Name != "پفک")
            {
                ModelState.AddModelError("", "لطفا مشکلات را برطرف کنید");
                ModelState.AddModelError("Name", "فقط محصولی با نام پفک قابل ثبت است");
                return View(product);
            }
            // todo:save...
        }
    }
}
```

```

        return RedirectToAction("Index");
    }
}

```

در قسمت کنترلر نیز اتفاق خاصی نیفتاده و کارهای پایه فقط انجام شده؛ ضمن اینکه آمدم برای داشتن خطاهای سفارشی نام محصول را چک کردیم و گفتیم اگر نام محصول چیزی غیر از «پفک» بود، از سمت سرور خطایی را صادر کند و بگوید که فقط پفک قابل ثبت هست.

#### View برنامه

```

@model FormValidationWithBootstrap.Models.ProductModel
@{
    ViewBag.Title = "New";
}
<h2>کالای جدید</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "alert alert-danger" })
        <div>
            @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2"
        })
            <div>
                @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })
            </div>
        </div>
        <div>
            @Html.LabelFor(model => model.Price, htmlAttributes: new { @class = "control-label col-md-2" })
            <div>
                @Html.EditorFor(model => model.Price, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Price, "", new { @class = "text-danger" })
            </div>
        </div>
        <div>
            @Html.LabelFor(model => model.Qty, htmlAttributes: new { @class = "control-label col-md-2"
        })
            <div>
                @Html.EditorFor(model => model.Qty, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Qty, "", new { @class = "text-danger" })
            </div>
        </div>
        <div>
            <div>
                <input type="submit" value="ثبت" />
                <input type="reset" value="ریست" />
                @Html.ActionLink("بازگشت به لیست", "Index", "Product", null, new { @class="btn btn-default" })
            </div>
        </div>
    </div>
}

```

فایل View برنامه با Scaffold Template ساخته شده و چون از Visual Studio 2013 استفاده شده، به صورت پیش‌فرض با بوت‌استرپ سازگار هست. تغییری که ایجاد شده تعویض کلاس مربوط به ValidationSummary هست که به alert alert-danger تغییر پیدا کرده و همچنین دو دکمه «ریست» و «بازگشت به لیست» هم به کنار دکمه «ثبت» اضافه شده.



## کالای جدید

<input type="text"/>	نام کالا
نام کالا یک فیلد اجباری است و باید آن را وارد کنید	
<input type="text"/>	قیمت
قیمت یک فیلد اجباری است و باید آن را وارد کنید	
<input type="text"/>	موجودی
موجودی یک فیلد اجباری است و باید آن را وارد کنید	
<input type="button" value="بازگشت به لیست"/>	<input type="button" value="ریست"/> <input type="button" value="ثبت"/>

در فرم بالا شاهد هستیم که با کلیک بر روی دکمه ثبت تنها خطاهای مرتبط با هر ردیف ظاهر شده‌اند و هیچ تغییر رنگی که حاصل از کلاس‌های مرتبط با Bootstrap باشند حاصل نشده. برای رفع این مشکل کافی‌است اسکریپت زیر، به انتهای فایل View برنامه اضافه شود تا پیش‌فرض‌های jQuery Validator را تغییر دهیم و آن‌ها را با بوت‌استرپ سازگار کنیم. همچنین در حالت ارسال فرم به سرور و Postback و نمایش خطاهای سفارشی، قسمت بررسی field-validation-error صورت می‌گیرد و در صورتیکه موردی را پیدا کند، به سطر مرتبط با آن کلاس has-error اضافه خواهد شد.

```
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
    <script>
        // override jquery validate plugin defaults
        $.validator.setDefaults({
            highlight: function (element) {
                $(element).closest('.form-group').addClass('has-error');
            },
            unhighlight: function (element) {
                $(element).closest('.form-group').removeClass('has-error').addClass('has-success');
            },
            errorElement: 'span',
            errorClass: 'help-block',
            errorPlacement: function (error, element) {
                if (element.parent('.input-group').length) {
                    error.insertAfter(element.parent());
                } else {
                    error.insertAfter(element);
                }
            }
        });
        $(function () {
            $('form').each(function () {
                $(this).find('div.form-group').each(function () {
                    if ($(this).find('span.field-validation-error').length > 0) {
                        $(this).addClass('has-error');
                    }
                });
            });
        });
    </script>
}
```

با افزودن اسکریپت فوق، در حالت اعتبارسنجی فرم‌ها به شکل زیر می‌رسیم:

## کالای جدید

<input type="text"/>	نام کالا
نام کالا یک فیلد اجباری است و باید آن را وارد کنید	
<input type="text"/>	قیمت
قیمت یک فیلد اجباری است و باید آن را وارد کنید	
<input type="text"/>	موجودی
موجودی یک فیلد اجباری است و باید آن را وارد کنید	
<input type="button" value="بازگشت به لیست"/>	<input type="button" value="ریست"/> <input type="button" value="ثبت"/>

همچنین هنگامیکه کاربر فیلد را به درستی وارد کرد، رنگ فیلد و همچنین آن ردیف به سبز تغییر خواهد کرد.

## کالای جدید

<input type="text" value="تست"/>	نام کالا
قیمت	
قیمت یک فیلد اجباری است و باید آن را وارد کنید	
<input type="text"/>	موجودی
موجودی یک فیلد اجباری است و باید آن را وارد کنید	
<input type="button" value="بازگشت به لیست"/>	<input type="button" value="ریست"/> <input type="button" value="ثبت"/>

و همچنین در حالت رخداد یک خطای سفارشی پس از postback از سمت سرور به حالت زیر خواهیم رسید.

## کالای جدید

• لطفا مشکلات را برطرف کنید!

نام کالا	<input type="text" value="آدامس"/>
	فقط محصولی با نام پفک قابل ثبت است :-)
قیمت	<input type="text" value="۱۲۰۰۰"/>
موجودی	<input type="text" value="۱۰"/>
	<div><div>ثبت</div><div>ریست</div><div>بازگشت به لیست</div></div>

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[FormValidationWithBootstrap.rar](#)

## نظرات خوانندگان

نویسنده: مهدی پایروند  
تاریخ: ۱۳۹۳/۰۵/۱۰ ۲۲:۳۴

سلام لطفا مثال پست را هم برای دانلود بذارید. ممنون

نویسنده: علی اکبر کورش فر  
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۳:۳۳

سلام. متأسفانه حجم پروژه به دلیل فایل‌های دی‌ال‌ال موجود 18 مگ بود و من نتونستم کمترش کنم

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۳:۳۸

برای ارسال پروژه‌ها در وب، همیشه پوشه‌های **packages**، پوشه‌های **bin** و **obj** را حذف کنید. نیازی به آن‌ها نیست، چون [قابل بازسازی](#) هستند.

نویسنده: علی اکبر کورش فر  
تاریخ: ۱۳۹۳/۰۵/۱۱ ۲۳:۰۰

خیلی ممنونم. خیلی دنبال این مسئله بودم. مچکرم از راهنماییتون

نویسنده: علی اکبر کورش فر  
تاریخ: ۱۳۹۳/۰۵/۱۱ ۲۳:۰۱

با عرض سلام. پروژه پیوست شد

نویسنده: هادی ام  
تاریخ: ۱۳۹۳/۰۵/۱۳ ۱۹:۲۱

این بوتسترپ شما، فارسی شده هست؟

نویسنده: علی اکبر کورش فر  
تاریخ: ۱۳۹۳/۰۵/۱۴ ۸:۲۸

فارسی‌شده در حد نیاز... قسمت‌هایی که ازشون استفاده می‌کنم رو فارسی کردم.  
فکر می‌کنم نسخه فارسی سازی شده بوت‌استرپ 3 هم هستش

نویسنده: محمد دلیری  
تاریخ: ۱۳۹۳/۰۵/۲۰ ۰:۵۱

اون قسمت‌های که میگی کجاها واسه فارسی کردن تغییر دادین که راست به چپ شده ؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۵/۲۰ ۱:۳۲

- [نسخه 3 راست به چپ](#)

- [نسخه 3.1.1 راست به چپ](#)

- [نسخه 3.2 راست به چپ](#)