

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 فایل‌های 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_rjl2U1nZrmSsaUyxAc1" rel="stylesheet"/>
<script src="/Scripts/js?v=GezdoTDiWY3acc3mI2Ujm_7nKKzh6Lu1Wr8TGyyLpW41"></script>
</head>
```

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

مفاهیم پایه‌ای Twitter Bootstrap

الف) Semantic class names

به عبارتی کلاس‌های Twitter Bootstrap دارای نام‌هایی معنا دار و مفهومی می‌باشند؛ مانند کلاس‌های CSS ای، به نام‌های Success، Error، Info و امثال آن. این نام‌ها مفهومی را می‌رسانند؛ اما در مورد نحوه پیاده سازی آن‌ها جزئیاتی را بیان نمی‌کنند. برای نمونه می‌توان کلاسی را به نام redText ایجاد کرد. هر چند این نام، توضیحاتی را در مورد علت وجودی‌اش بیان می‌کند، اما بسیار ویژه بوده و در مورد جزئیات پیاده سازی آن نیز اطلاعاتی را ارائه می‌دهد. 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 نام‌های مفهومی یاده شده، مانند Error استفاده می‌شود. نام‌هایی معنا دار اما بدون دقیق شدن در مورد ریز جزئیات پیاده سازی آن‌ها. در این حالت می‌توان [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 می‌تواند sidebar باشد تا بدون دقیق شدن در جزئیات پیاده سازی آن، در چپ یا راست صفحه قابل اعمال باشد. Semantic class names کلیدهای هستند جهت استفاده مجدد از قابلیت‌های یک فریم ورک CSS.

ب) Compositional classes

اکثر کلاس‌های Twitter Bootstrap دارای محدوده کاری کوچکی هستند و به سادگی قابل ترکیب با یکدیگر جهت رسیدن به نمایی خاص می‌باشند. برای مثال به سادگی می‌توان 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 کلاس‌های 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-striped table-hover 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>

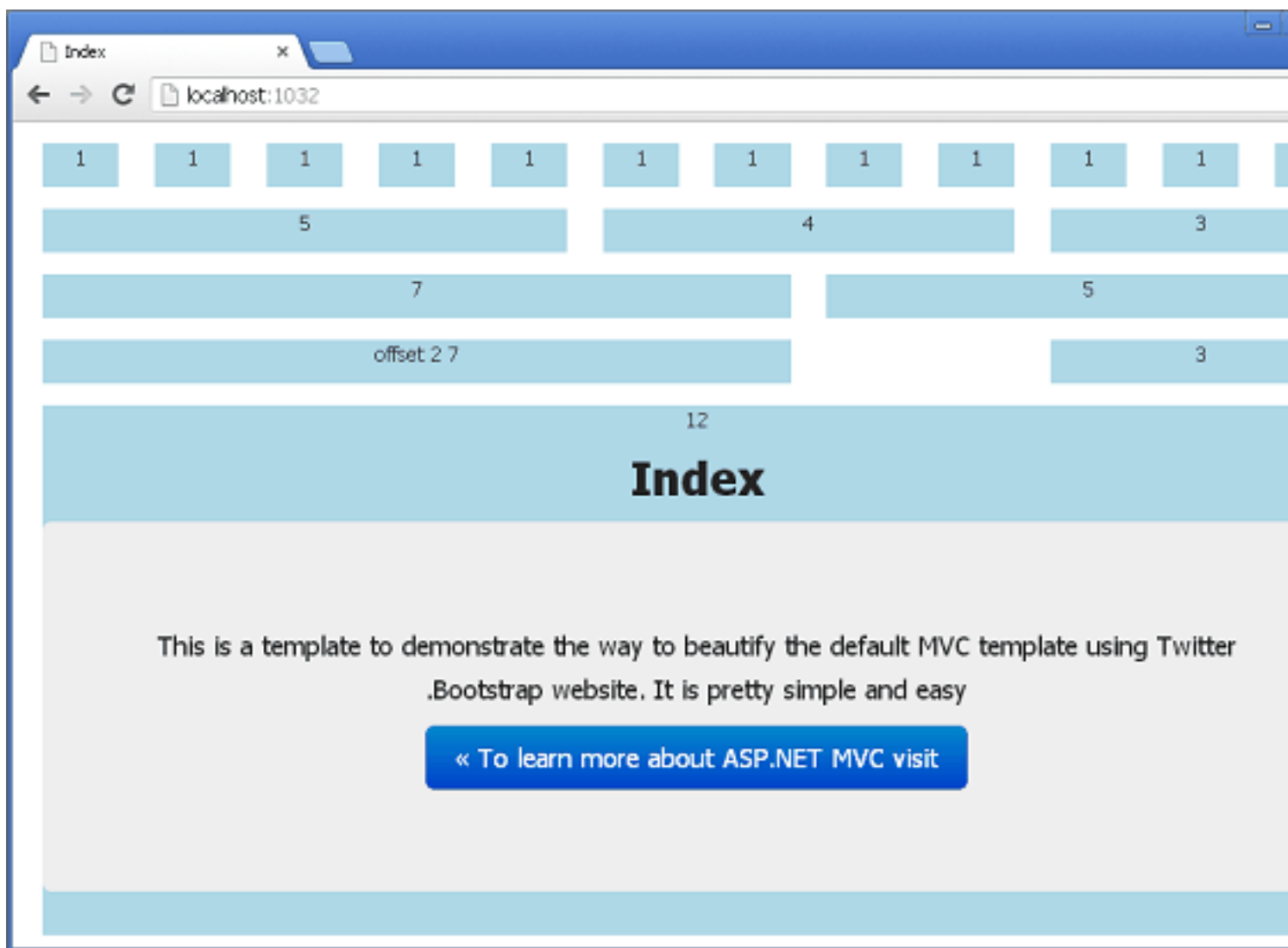
```

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>
```

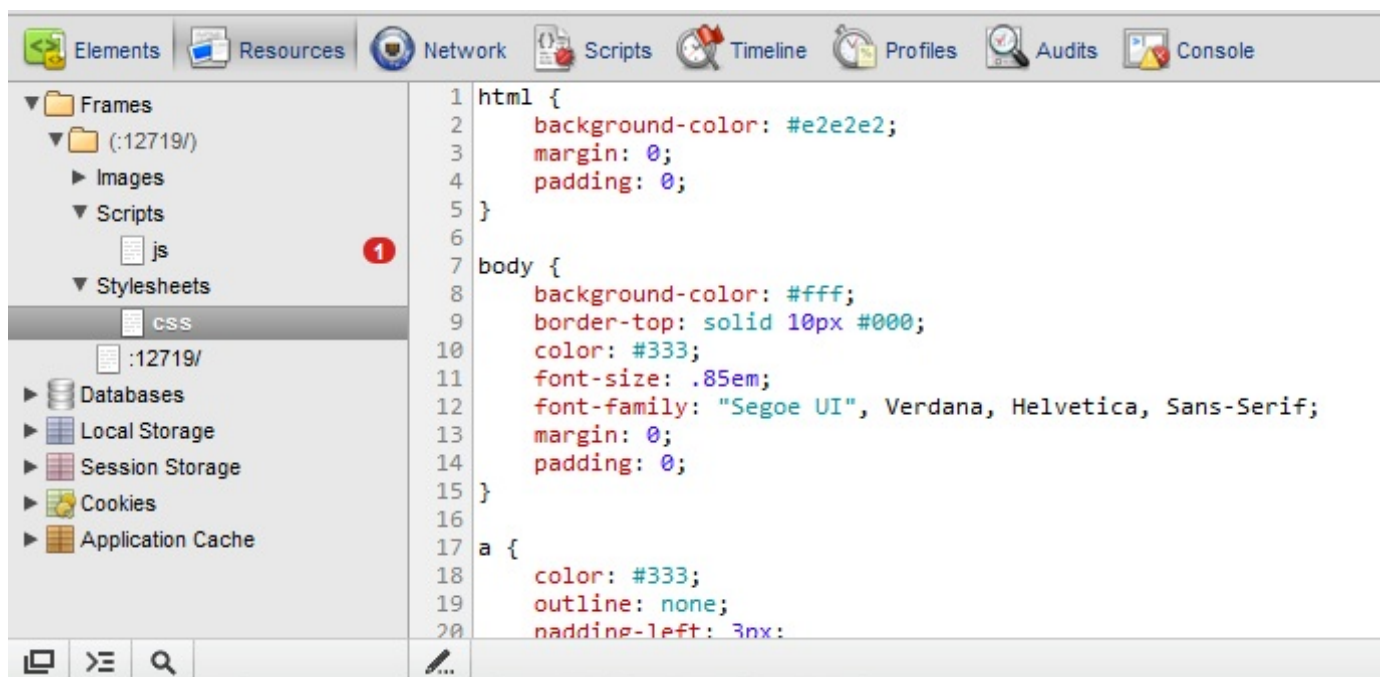
منتها برای CSS فقط فایل Style.css و برای جاوااسکریپت فقط jQuery رو اضافه میکنه، به نظرتون مشکل چی میتونه باشه؟

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

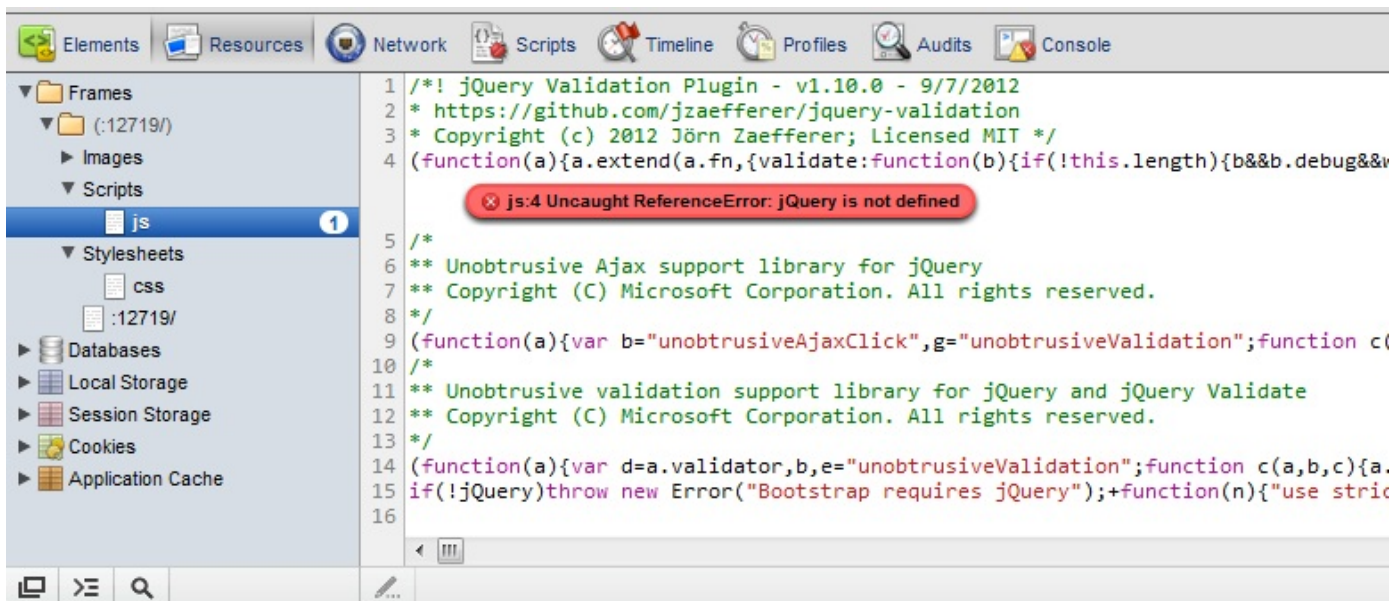
اگر واقعا مشکلی وجود داشته باشد، در همین لینک‌های پویایی که ارائه دادید، با کلیک بر روی آن‌ها عبارت «Minification failed. Returning unminified contents» ظاهر خواهد شد ([یک نمونه مثال در اینجا](#)) و سپس در سطرهای بعدی علت خطای کامپایل ذکر می‌شود که باید برطرف شود.

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 می‌خواه 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 بوت‌استرپ شدم.

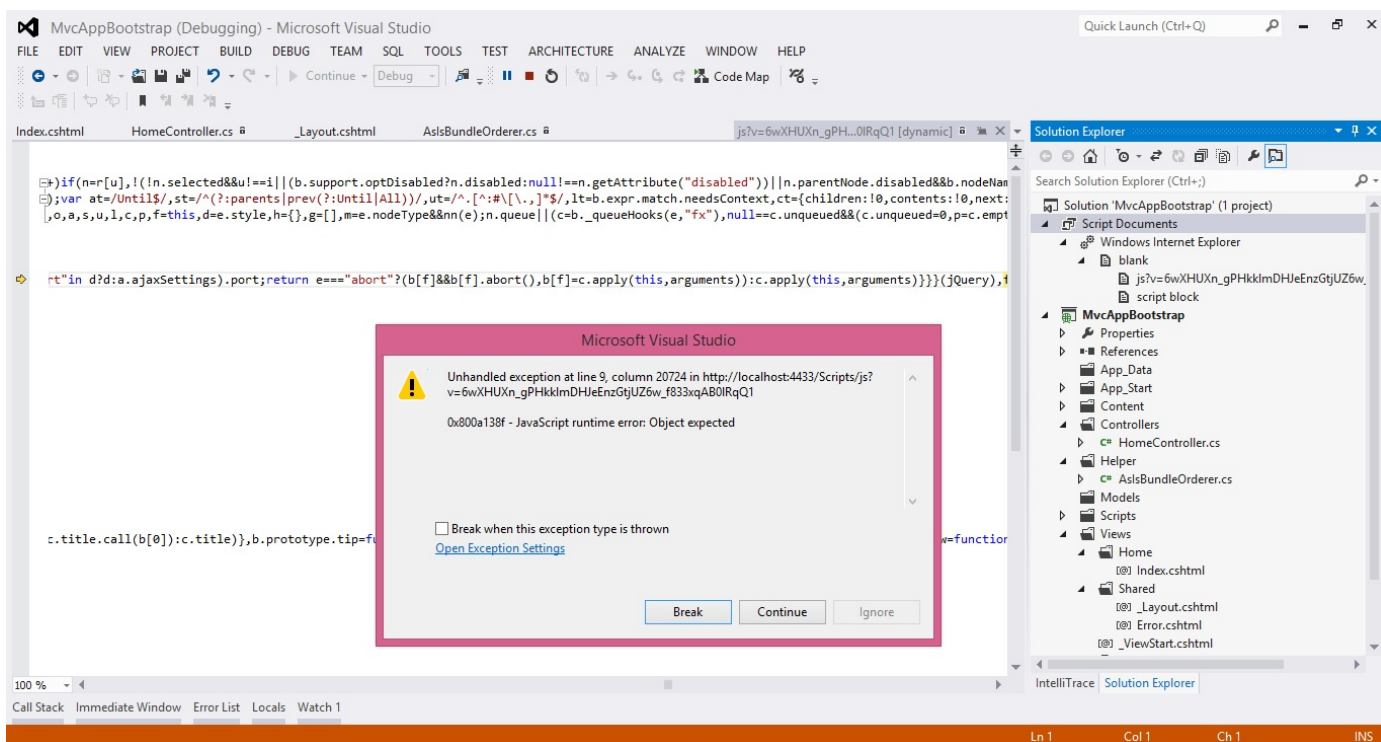
نویسنده: وحید نصیری
تاریخ: ۱۷:۴۵ ۱۳۹۲/۰۹/۲۰

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 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](#).

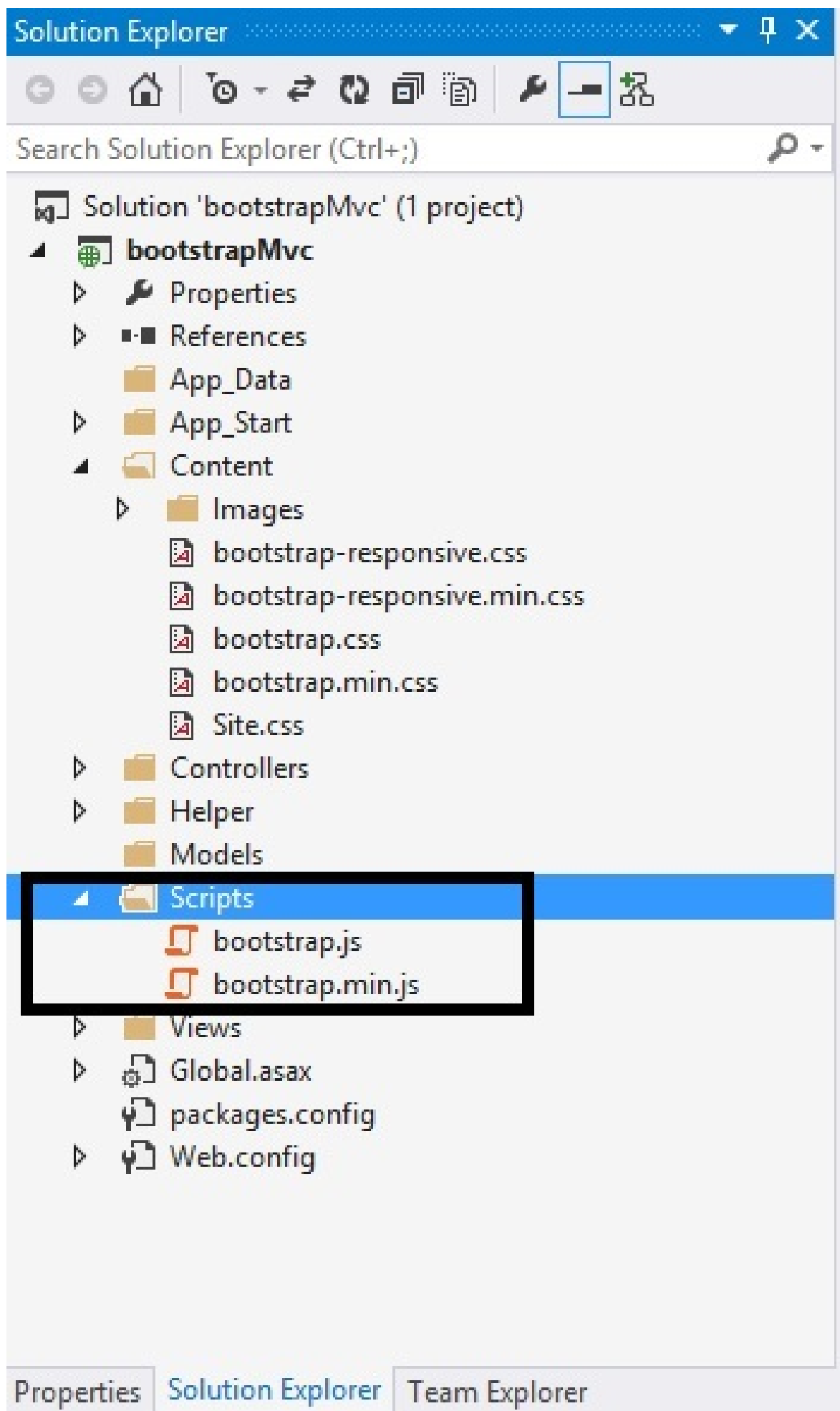
نویسنده: ایمان باقری
تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۳:۲۵

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 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:

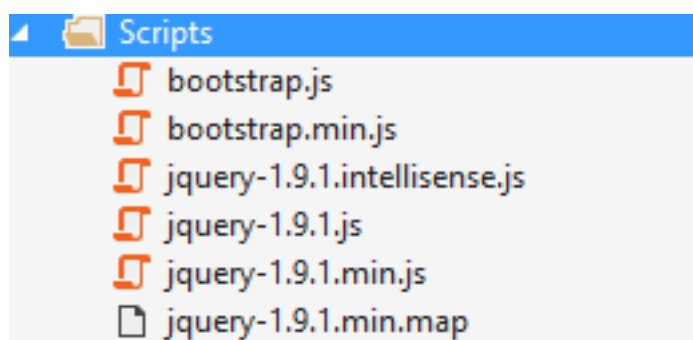
ایمان باقری

تاریخ:

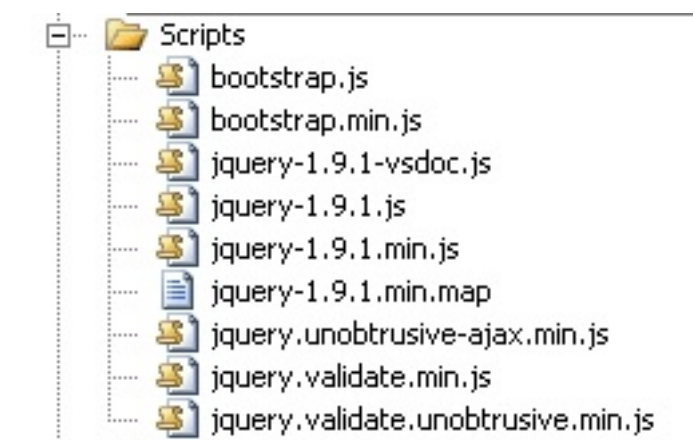
۱۴:۳ ۱۳۹۳/۰۱/۱۹

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 !

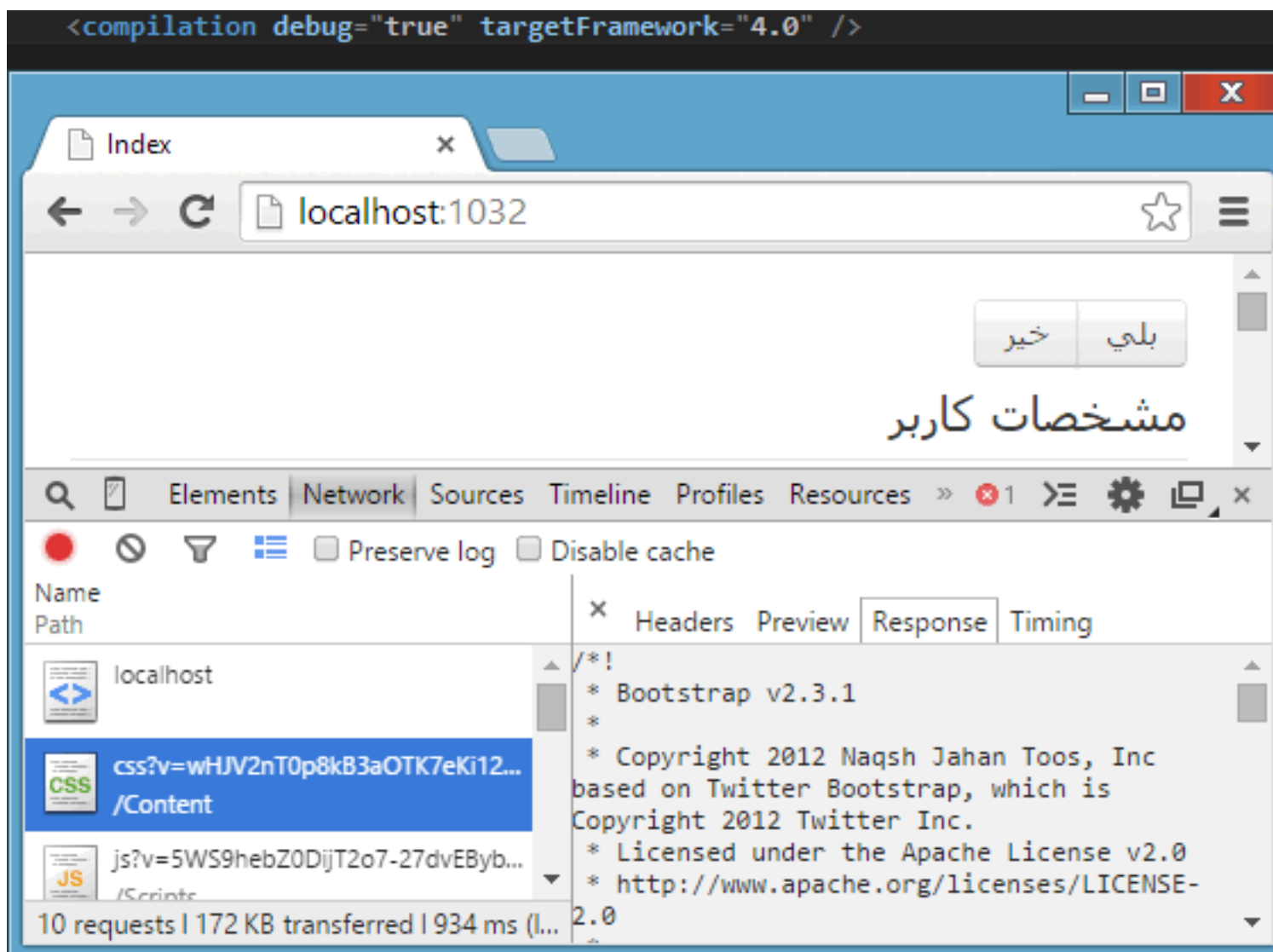
نویسنده: وحید نصیری

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.

نویسنده: امیرتنها
تاریخ: ۲۳:۱۷ ۱۳۹۳/۱۰/۰۵

با سلام

هنگام اجرا این خطا رو می‌ده .

'does not contain a definition for 'AddStyles

Line 19: @BundleConfig.AddStyles("~/Content/css",

Line 20: "~/Content/bootstrap.css",

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۵ ۲۳:۴۵

یکبار نظرات این بحث را کامل مطالعه کنید. این مطلب برای بوت استرپ 2 نوشته شد، در آن زمان. نگارش 3 آن [در اینجا](#) بحث شده. همچنین این کلاسها [تغییراتی داشته‌اند](#) که پیشتر در نظرات بحث شده.

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۲/۲۸ ۱۸:۵

یک مشکلی که من با این کد بهش برخوردم این بود که اگر اسکریپتی به وابستگی به اسکریپت‌های دیگه داشته باشه ممکن هست دچار مشکل بشه
برای مثال جی کوئری و پلاگین هاش
به نظر میرسه ترتیب معرفی اسکریپت‌ها رو رعایت نمی‌کنه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۲/۲۹ ۲۱:۱۵

- AsIsBundleOrderer به همین دلیل تهیه شده‌است تا ترتیب رعایت شود.
- برای دیباگ آن BundleTable.EnableOptimizations را false کنید و همچنین برنامه را در حالت دیباگ در ویژوال استودیو اجرا کنید. در این حالت لینک نهایی تولید شده‌ی توسط آن، هرچند یک لینک است، اما خروجی آن فشرده نشده‌است و می‌توانید در آن ترتیب قرار گرفتن اسکریپت‌ها را بهتر بررسی کنید.

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><div class="row-fluid"> <div class="span12"> <p class="muted"> متن غیرفعال <p class="text-warning"> نمایش اخطار به کاربر <p class="text-error"> نمایش خطا به کاربر <p class="text-info"> نمایش اطلاعات به کاربر <p class="text-success"> نمایش موفقیت آمیز بودن عملیات </div> </div></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><div class="row-fluid"> <div class="span12"> <p> My <abbr title="منظور واحد پردازش مرکزی است"> CPU</abbr> has N Cores. </p> </div> </div></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 نیست وجود دارد (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>

```

```
<label>نام:</label>
<input name="name" type="text"
/>
</fieldset>
</form>
</div>
</div>
</div>
```

در اینجا یک فرم ساده را مشاهده می‌کنید. بدون bootstrap، تمام عناصر این فرم در یک سطر نمایش داده می‌شوند. اما با صرفاً فعال سازی css مرتبط با آن، به شکل مدرن فوق خواهیم رسید. همانطور که ملاحظه می‌کنید، کلیه عناصر را به صورت یک پشته عمودی در صفحه قرار داده‌است و نکته مهم اینجا است که هیچگونه شیوه نامه خاص و اضافه‌تری به فرم فوق از طرف ما اعمال نشده است.

نکته: اگر می‌خواهید همان حالت پیش فرض مرورگر، یعنی قرار دادن تمام عناصر در پشت سر هم را فعال کنید، تنها کافی است کلاس form-inline را به form تعریف شده فوق اضافه نمایید.

ثبت نام

ایمیل: | نام:

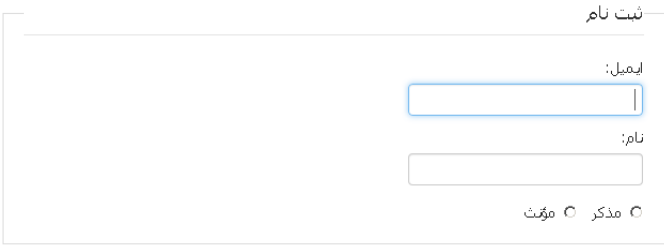
2) نکاتی در مورد checkboxes و radio buttons

در حالت پیش فرض، با تعریف یک label و checkbox، برچسب متناظر با آن اندکی بالاتر از checkbox قرار گرفته شده در صفحه ظاهر می‌شود. برای رفع این مشکل تنها کافی است کلاس checkbox به label اعمال شود (و برای radio button از کلاس radio استفاده خواهد شد):

```
<label class="checkbox">
  <input type="checkbox" name="isMale" />
  مذكر
</label>
```

این مثال را یکبار با کلاس checkbox و بار دیگر بدون آن آزمایش کنید تا تفاوت را بهتر بتوان درک کرد.


نکته: برای قرار دادن چندین checkbox یا radio button در یک سطر (با توجه به حالت چیدمان عمودی پیش فرض فرم‌ها)، ابتدا آن‌ها را داخل یک div قرار دهید. سپس به تمام checkboxها یا radio buttonها کلاس inline را نیز اضافه نمایید. برای مثال:

<pre> <div> <label class="radio" inline"> <input type="radio" name="isMale" /> مذکر</label> <label class="radio" inline"> <input type="radio" name="isFemale" /> مؤنث</label> </div> </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> <form class="search-form"> <fieldset> <legend>جستجو</legend> <input type="search" class="search-query" /> <button type="submit" class="btn" >بیاب</button> </fieldset> </form> </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> <form class="form-horizontal" action="/signup" method="post"> <fieldset> </pre>	
---	--

<pre> <legend>ثبت نام</legend> <div class="control-group"> <label class="control- label"> ایمیل:</label> <div class="controls"> <input name="email" type="text" /> </div> </div> <div class="control-group"> <label class="control- label"> نام:</label> <div class="controls"> <input name="name" type="text" /> </div> </div> <div class="control-group"> <div class="controls"> <label class="radio inline"> <input type="radio" name="isMale" /> مذکر</label> <label class="radio inline"> <input type="radio" name="isFemale" /> مؤنث</label> </div> </div> </div> </form> </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> <div class="control-group error"> <label class="control-label"> ایمیل:</label> <div class="controls"> <input name="email" type="text" /> لطفاً ایمیل را با فرمت صحیحی وارد نمایید</div> </div> </div> </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><div class="input-prepend input-append"> .00 <input dir="ltr" type="text" /> ریال </div></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</code>
</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 حتماً 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 نمایید.

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 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 نمایید؛ تا شروع مدخل مرتبط با آیکون‌ها را بتوانید مشاهده نمایید. 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 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><div class="input-prepend"> <i class="icon-envelope"></i> <input type="email" dir="ltr" /> </div></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 طرح بندی](#) 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 افزارای 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.

نویسنده: امیر صیدی لو
تاریخ: ۱۹:۳۲ ۱۳۹۴/۰۳/۲۷

در 2012 vs هنگام استفاده از کلاسهای css و همچنین Twitter Bootstrap ، intellisense کار نمیکنه آیا راهی هست؟

نویسنده: وحید نصیری
تاریخ: ۱۹:۳۸ ۱۳۹۴/۰۳/۲۷

افزونهی [Web Essentials](#) را نصب کنید.

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

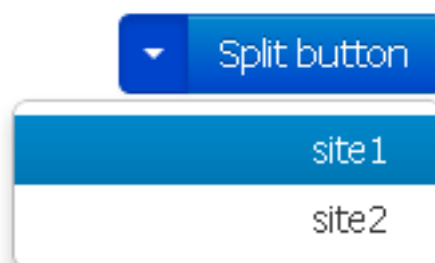
نیز نسبت داد. به این ترتیب، منو از محل قرارگیری این المان باز خواهد شد. در ادامه روال متداول تعریف منو را در Bootstrap، ملاحظه می‌کنید. کلاس dropdown-toggle سبب خواهد شد تا با کلیک ماوس بر روی لینک، منو ظاهر شود. کلاس caret نیز سبب نمایش مثلث رو به پایین، جهت مشخص سازی وجود منو در این محل خاص، تعریف 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 راهنمایی 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

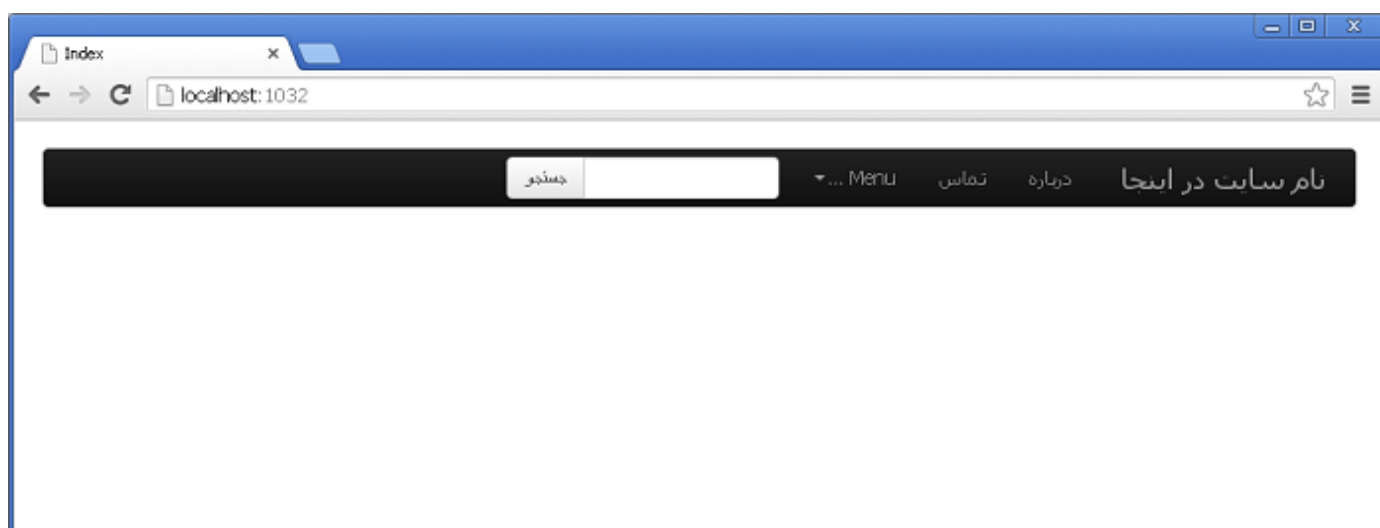
اگر علاقمند باشید که منوی راهبری بالای سایت دارای یک حاشیه به همراه مسایل طراحی واکنش‌گرا باشد، می‌توان از کامپوننت navbar مجموعه Bootstrap استفاده کرد:

```
<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.

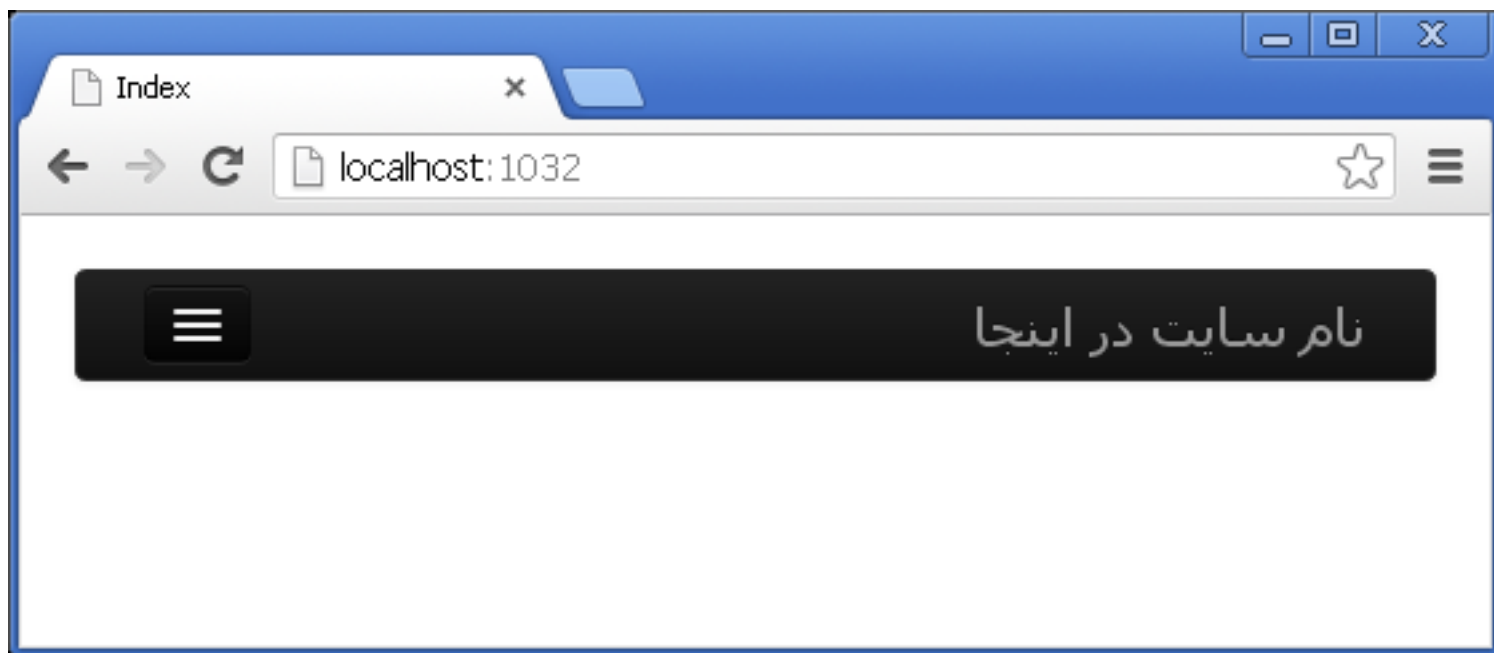
عموماً مرسوم است که در منوی راهنبری اصلی سایت، نام سایت یا متنی مشخص نیز در قسمتی از این منو ظاهر شود. برای انجام اینکار، کافی است یک لینک را با کلاس brand اضافه کنیم. این موارد، اصول کلی تهیه یک منوی راهنبری اصلی سایت را توسط Bootstrap بیان می‌کنند. در ادامه به چند نکته تکمیلی خواهیم پرداخت:

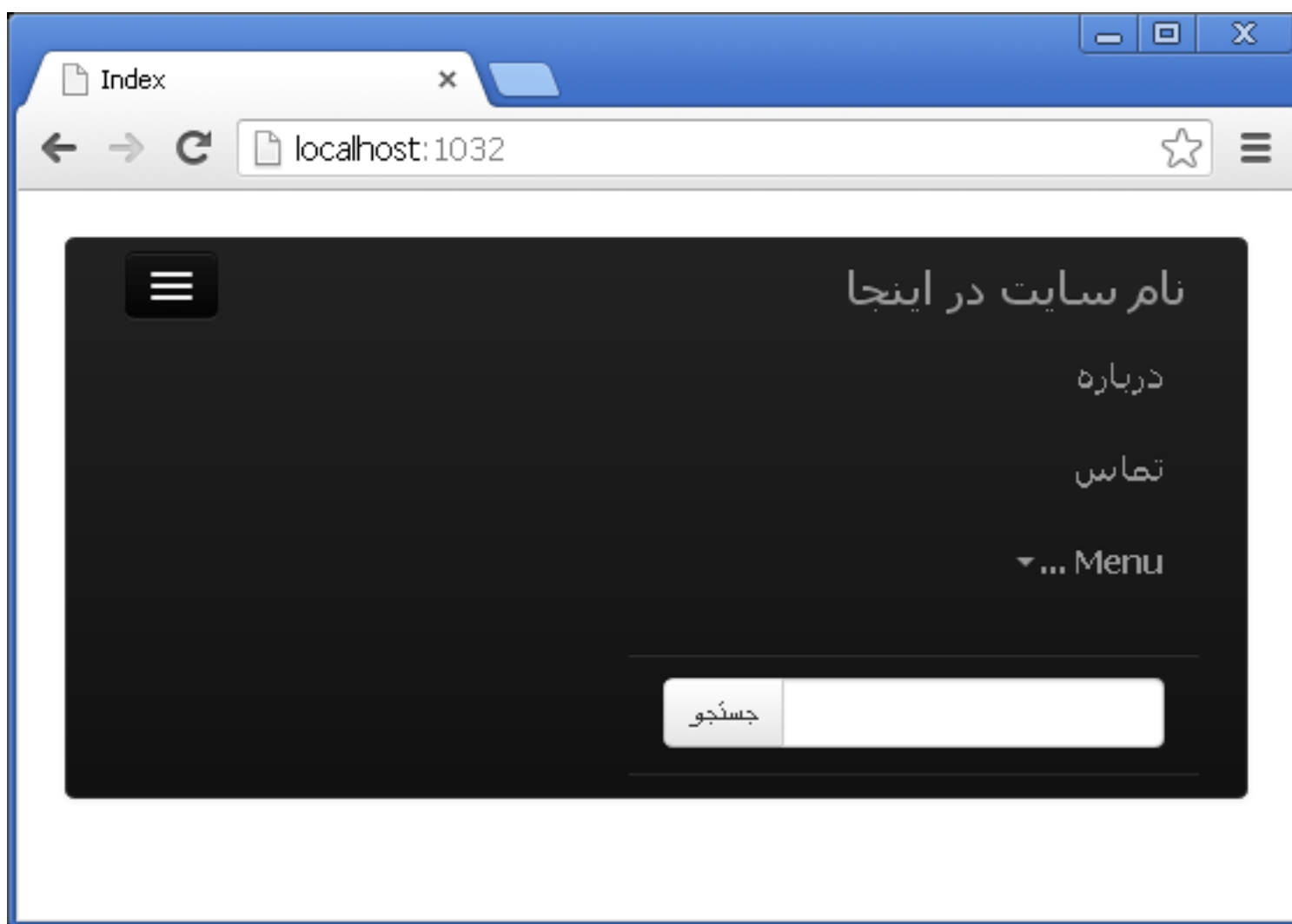
- اگر به تگ nav، کلاس navbar-static-top را اضافه کنیم، فاصله بین منو و اطراف صفحه را حذف می‌کند.
- اگر علاقمند هستیم که منوی بالای صفحه حتی با اسکرول به پایین محتوای نمایش داده شده، همواره نمایان باشد و بر روی

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>
```

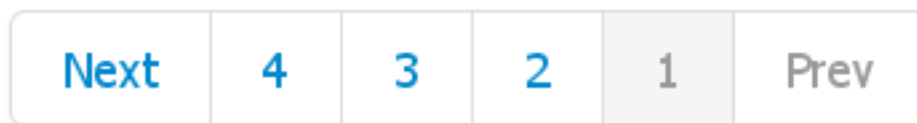
سطح يك / سطح دو / سطح سه

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

خطار

مهم است

اطلاعات

رنگ معکوس

پیش فرض

موفقیت آمیز

خطار

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 خطار 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](#)

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.

نویسنده: روناک تابعی
تاریخ: ۲۳:۴۷ ۱۳۹۴/۰۵/۱۱

کلاس‌های bootstrap ذکر شده مربوط به نسخه 2 است که بیشترشون در نسخه 3 تغییر کردن. برای مثال `alert-error` به `alert-danger` تغییر کرده. [Migrating from 2.x to 3.0](#)

نویسنده: وحید نصیری
تاریخ: ۲۳:۵۲ ۱۳۹۴/۰۵/۱۱

- « [ارتقاء به Bootstrap 3](#) »

- « [سازگاری کلاس‌های اعتبارسنجی Twitter Bootstrap 3 با فرم‌های ASP.NET MVC](#) »

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("ResourceName") })
        </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](#) »

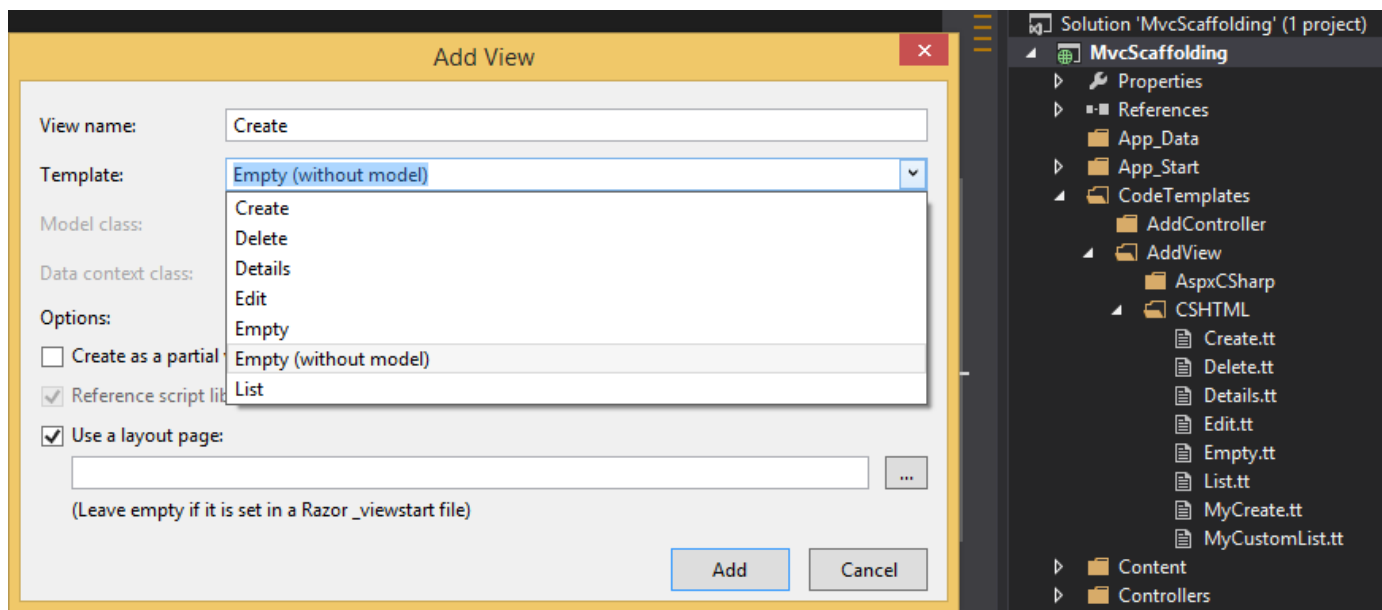
نویسنده: محمد صاحب
تاریخ: ۱۶:۵۶ ۱۳۹۲/۱۲/۱۰

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>

با تعدادی از کامپوننت‌های Bootstrap در مطلب « [نگاهی به اجزای تعاملی Twitter Bootstrap](#) » آشنا شدید. یکی دیگر از این افزونه‌ها، Typeahead نام دارد که در حقیقت نوعی Autocomplete text box است. در ادامه قصد داریم نحوه استفاده از آن را در ASP.NET MVC بررسی کنیم.

استفاده‌ی استاتیک از افزونه Typeahead

منظور از استفاده‌ی استاتیک، مشخص بودن آرایه عناصر و همچنین درج آن به صورت html encoded در صفحه است. برای این منظور، کنترلر برنامه چنین شکلی را خواهد داشت:

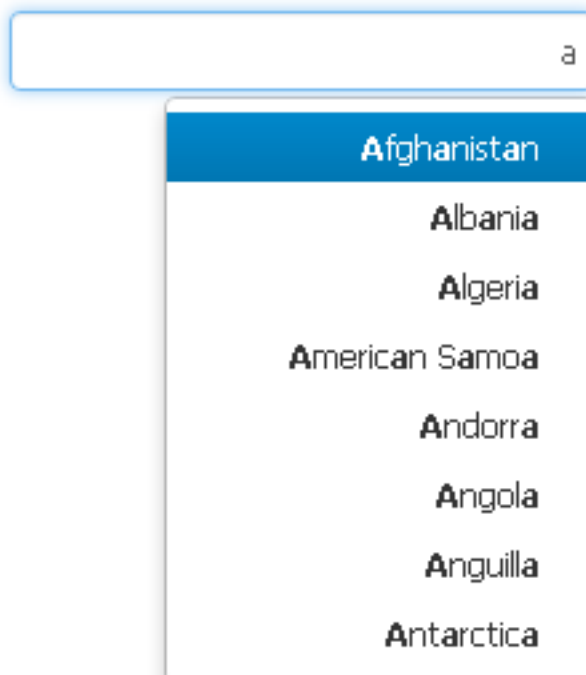
```
using System.Web.Mvc;
using System.Web.Script.Serialization;

namespace Mvc4TwitterBootstrapTest.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            var array = new[]
            {
                "Afghanistan",
                "Albania",
                "Algeria",
                "American Samoa",
                "Andorra",
                "Angola",
                "Anguilla",
                "Antarctica",
                "Antigua and/or Barbuda"
            };
            ViewBag.JsonString = new JavaScriptSerializer().Serialize(array);
            return View();
        }
    }
}
```

در اینجا یک آرایه با تعداد عناصر مشخص، تبدیل به رشته JSON معادل آن شده و توسط ViewBag.JsonString به View ارسال می‌شود. View متناظر با آن به نحو ذیل با مشخص سازی نوع data-provide (تا به کتابخانه‌ی جاوا اسکریپتی همراه bootstrap اعلام کند از چه افزونه‌ای در اینجا قرار است استفاده شود)، منبع داده data-source و حداکثر تعداد آیتم ظاهر شونده data-items، می‌تواند طراحی شود:

```
@{
    ViewBag.Title = "Index";
}
<h2>
    Typeahead</h2>
@Html.TextBox("search", null, htmlAttributes:
    new
    {
        autocomplete = "off",
        data_provide = "typeahead",
        data_items = 8,
        data_source = @ViewBag.JsonString
    })
```

Typeahead



به این ترتیب، یک چنین خروجی در صفحه درج می‌شود:

```
<input autocomplete="off" data-items="8" data-provide="typeahead"
data-source="['"Afghanistan"','"Albania"','"Algeria"','"American
Samoa"','"Andorra"','"Angola"','"Anguilla"','"Antarctica"','"An
tigua and/or Barbuda"']"
id="search" name="search" type="text" value="" />
```

همانطور که ملاحظه می‌کنید دقیقاً data-source تهیه شده مطابق نیاز خاص این افزونه، html encoded است. به علاوه هر جایی در htmlAttributes صفحه از under line استفاده شده، در این سمت به صورت خودکار به - ترجمه گردیده است. اگر هم بخواهیم برای آن یک Html Helper درست کنیم، می‌توان به نحو ذیل عمل کرد:

```
public static MvcHtmlString TypeaheadFor<TModel, TValue>(
    this HtmlHelper<TModel> htmlHelper,
    Expression<Func<TModel, TValue>> expression,
    IEnumerable<string> source,
    int items = 8)
{
    var jsonString = new JavaScriptSerializer().Serialize(source);
    return htmlHelper.TextBoxFor(
        expression,
        new
        {
            autocomplete = "off",
            data_provide = "typeahead",
            data_items = items,
            data_source = jsonString
        }
    );
}
```

استفاده پویا و Ajax ایی از افزونه Typeahead

اگر بخواهیم data-source را به صورت پویا، هربار از بانک اطلاعاتی دریافت و ارائه دهیم، نیاز به کمی اسکریپت نویسی خواهد بود:

```
using System;
using System.Linq;
using System.Web.Mvc;
using System.Web.Script.Serialization;

namespace Mvc4TwitterBootstrapTest.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public JsonResult GetNames(string term)
        {
            var array = new[]
            {
                "Afghanistan",
                "Albania",
                "Algeria",
                "American Samoa",
                "Andorra",
                "Angola",
                "Anguilla",
                "Antarctica",
                "Antigua and/or Barbuda"
            };

            var results = array.Where(n =>
                n.StartsWith(term, StringComparison.OrdinalIgnoreCase));

            return Json(results.ToArray(), JsonRequestBehavior.AllowGet);
        }
    }
}
```

در این حالت، کدهای اکشن متدی که یک عبارت، یا قسمتی از آن را از طریق پارامتر term دریافت و خروجی JSON مناسبی را ارائه می‌کند، همانند متد GetNames فوق خواهد بود.

سپس در تعاریف View، قسمت data-source مرتبط با TextBox حذف و از طریق فراخوانی مستقیم کدهای افزونه typeahead مقدار دهی می‌گردد:

```
@{
    ViewBag.Title = "Index";
    var url = Url.Action("GetNames", "Home");
}
<h2>
    Typeahead</h2>
@Html.TextBox("search", null, htmlAttributes:
    new
    {
        autocomplete = "off",
        data_provide = "typeahead",
        data_items = 8
    })
@section JavaScript
{
    <script type="text/javascript">
        $(function () {
            $('#search').typeahead({
                source: function (term, process) {
                    return $.getJSON('@url', { term: term }, function (data) { return
process(data); });
                }
            });
        });
    </script>
}
```


در اینجا توسط متد getJSON کتابخانه jQuery، مقدار عبارت وارد شده در TextBox جستجو، به آدرس اکشن متد GetNames ارسال و سپس حاصل به source افزونه typeahead انتساب داده می‌شود.

نظرات خوانندگان

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۲/۰۶/۲۵ ۱۰:۴۵

آیا پیاده سازی تابع matcher افزونه typeahead با استفاده از HtmlHelper امکان پذیر است؟

یا به عبارت دیگر: آیا امکان پیاده سازی کامل این افزونه در سمت سرور وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۲۵ ۱۱:۷

matcher یک callback جاوا اسکریپتی است. بنابراین در سمت کلاینت باید پیاده سازی شود (چیزی شبیه به مقدار دهی source پویای مثال آخر بحث).

```
$('.typeahead').typeahead({
  matcher: function(item) {
    // آیتم مقداری است که باید برای تطابق بررسی شود
    // کوئری جاری را بر می گرداند
    return true // اگر آیتم تطابق داشته است
  }
})
```

مثلا:

```
matcher: function (item) {
  if (item.toLowerCase().indexOf(this.query.trim().toLowerCase()) != -1) {
    return true;
  }
}
```

از matcher برای سفارشی سازی و بررسی اینکه آیا عبارت تایپ شده توسط کاربر با list دریافتی تطابق دارد یا خیر، استفاده می شود. حالت پیش فرض، تطابق دقیق لیست دریافتی با متن ورودی کاربر است. با استفاده از matcher در سمت کلاینت می توانید نحوه نمایش لیست دریافتی از سرور را سفارشی سازی کنید.

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۲/۰۶/۲۵ ۱۴:۴

نکته اضافی: در صورتی که بخواهید کاربر فقط مقادیر داخل لیست را وارد نماید، از کد زیر استفاده می کنیم:

```
matcher: function (item) {
  return true;
}
```

نویسنده: رهام
تاریخ: ۱۳۹۲/۰۹/۰۵ ۱۱:۲

سلام، وقت بخیر

از کد شما برای آزمایش Typeahead استفاده کردم، اما کار نمیکنه. هیچ پیام خطایی وجود نداره، فقط در TextBox نهایی autocomplete نداره. نمیدونم کجای کار رو اشتباه انجام دادم. در جستجو پیدا کردم که CSS هم برای Typeahead وجود داره که من در فایل bootstrap.css چیزی پیدا نکردم. برنامه بدون خطا اجرا میشه و باقی بخشها به درستی کار میکنه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۰۵ ۱۱:۳۱

مطلب فوق مربوط به بوت استرپ دو است. این افزونه در بوت استرپ سه حذف شده و باید از کتابخانه‌های ثالث استفاده کنید.
[برای نمونه](#)

نویسنده: رضا
تاریخ: ۱۸:۲۷ ۱۳۹۲/۰۹/۱۸

سلام ؛ تشکر از مطلب مفیدتون.

چند تا سوال در کار کردن با TypeAhead برام پیش اومد که هنوز نتونستم حلش کنم

- 1.چطور میشه نتایج را با یک قالب خاص نمایش داد (Template) نمایش تصویر ، گروه بندی و ... برای هر مورد پیشنهادی
 - 2.چطور می‌شه که هر مورد پیشنهادی یک لینک داشته باشد و با کلیک بر روی آن به لینک ارجاع داده شود و نه اینکه در کادر قرار گیرد (برای کادر جستجو)
- در ضمن هر دو مورد در پروژه IRIS Cms وجود داشت ولی نتونستم روش کارکردنش و بفهمم.

نویسنده: وحید نصیری
تاریخ: ۲۰:۴۳ ۱۳۹۲/۰۹/۱۸

IRIS از [Chosen](#) استفاده کرده؛ سایت جاری از [jQuery Autocomplete](#) .

نویسنده: رضا
تاریخ: ۱۰:۱۵ ۱۳۹۲/۰۹/۱۹

ممنون از پاسختون ولی من حتما باید با استفاده از Bootstrap typeahead این کار و بکنم ،در صورت امکان ممنون می‌شم راهنمای کنید .

نویسنده: وحید نصیری
تاریخ: ۱۰:۲۹ ۱۳۹۲/۰۹/۱۹

از highlighter آن باید استفاده کنید:

```
$('.typeahead').typeahead({  
  highlighter: function(item){  
    return "<div>.....</div>";  
  }  
});
```

در اینجا item را به صورت یک شیء کامل بازگشت داده شده توسط JSON تعریف کنید که مثلا آدرس یک URL یا یک ID و متن و غیره را داشته باشد. بعد در return آن تصویر قرار دهید یا لینک و یا هر پردازش دیگری که لازم است.

یک مثال کامل: [Heavily Customizing a Bootstrap Typeahead](#)

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="471 956 523 970" data-label="Page-Footer">
<p>۶۸/۱۷۶</p>
</div>
```

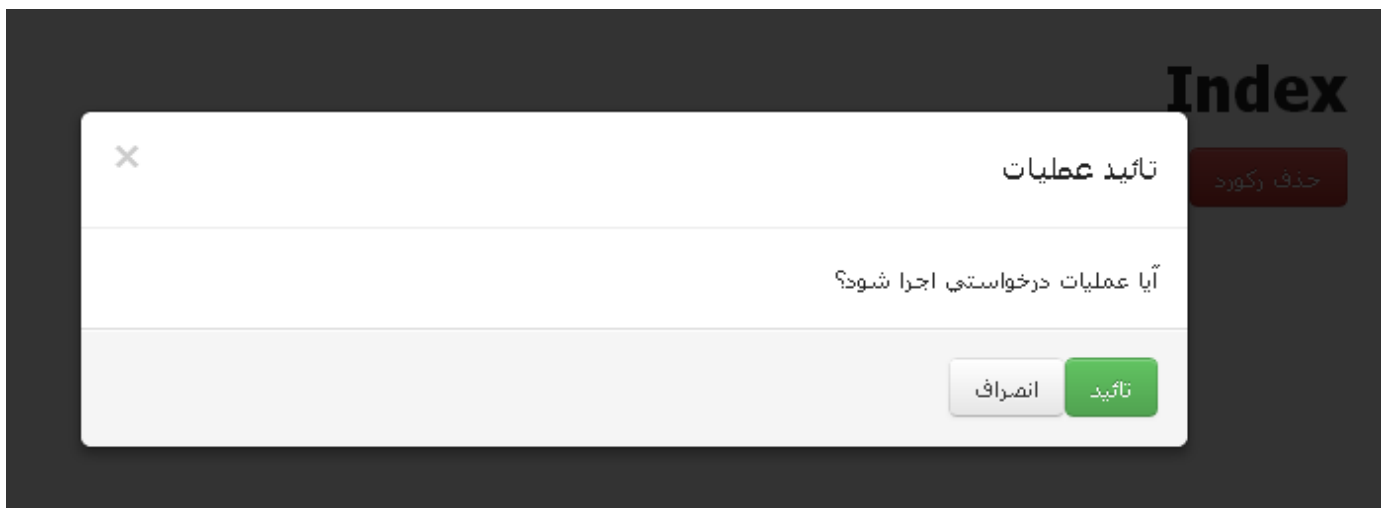
مثالی از استفاده از افزونه bootstrapModalConfirm

```
@{
    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">۷۳/۱۷۶</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="تکمیل کنید")]
        public string Name { set; get; }

        [DisplayName("نام خانوادگی")]
        [Required(ErrorMessage = "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;  
        returnJson(new{ success = true});  
    }  
  
    this.ModelState.AddModelError("", "خطایی رخ داده است");  
    returnPartialView("_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 این دو 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 }),
```

در این مطلب با نحوه استفاده از پلاگین جستجوی سفارشی searchboxmvc.js آشنا خواهید شد.



قبلاً در [اینجا](#) با نحوه ایجاد پلاگین jQuery آشنا شدید. روشی دیگری نیز برای ایجاد این نوع پلاگین‌ها وجود دارد و آن استفاده از [widget factory](#) موجود در پلاگین jQuery UI می‌باشد.

برای استفاده از این پلاگین که کدهای کامل آن در فایل پیوست موجود است، ابتدا باید فایل‌های لازم را به پروژه خود اضافه کنیم:

```
<link rel="stylesheet" href="@Url.Content("~/Content/bootstrap-rtl.css")" type="text/css" />
<script type="text/javascript" src="@Url.Content("~/scripts/jquery-2.0.2.min.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/jquery-ui-1.10.3.min.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/bootstrap-rtl.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/searchboxmvc.js")"></script>
```

سپس در کنترلر خود یک Action بصورت زیر ایجاد کنید:

```
[HttpPost]
public virtual ActionResult LoadData(string fieldName, string value, string stringFilterMode =
"startWith")
{
    Thread.Sleep(2000);
    var models = MakePersons();
    if (fieldName == "Id")
    {
        models = models.Where(p => p.Id == int.Parse(value)).Take(1).ToList();
    }
    else if (fieldName == "FirstName")
    {
        models = models.Where(p => p.FirstName.StartsWith(value)).ToList();
    }

    return Json(new { Status = "OK", Records = models });
}
private List<Person> MakePersons()
{
    var lst = new List<Person>();
    lst.Add(new Person() { Id = 1, Code = "Uytffs-098", FirstName = "احمد رضا", LastName =
"عابدزاده" });
    lst.Add(new Person() { Id = 2, Code = "fTuuw-652", FirstName = "کریم", LastName = "باقری"
});
    lst.Add(new Person() { Id = 3, Code = "Lopapo-123", FirstName = "خداداد", LastName =
"عزیزی" });
    lst.Add(new Person() { Id = 4, Code = "Utppq-981", FirstName = "علی", LastName = "دایی" });
    lst.Add(new Person() { Id = 5, Code = "zttsn-471", FirstName = "علی", LastName = "کریمی"
});
}
```

```
lst.Add(new Person() { Id = 6, Code = "poiud-901", FirstName = "مهدی", LastName = "مهدوی" });
lst.Add(new Person() { Id = 7, Code = "wqrPoP-391", FirstName = "علیرضا", LastName = "منصوریان" });
return lst;
}
```

در ادامه در ویوی مورد نظر خود یک div ایجاد کنید. همین div خام با اعمال پلاگین بر روی آن، بصورت یک پلاگین جستجو عمل خواهد کرد.

حال کدهای جاوا اسکریپت مورد نظر را برای اعمال پلاگین و تنظیمات مورد نیاز آن به div ایجاد شده می‌نویسیم:

```
<div id="div_SearchBoxContainer">
</div>
<script type="text/javascript">
$(function() {
    $('#div_SearchBoxContainer').searchboxmvc({
        loadUrl: '@Url.Action(actionName: "LoadData", controllerName: "Home")',
        defaultStringFilterMode: "startWith",
        loadDataOnLeave: true,
        displayClass: "",
        displayNoResultClass: "",
        display: function (element, record) {
            $(element).html(record.FirstName + " " + record.LastName);
        },
        listItemsDisplay: function (element, record, index) {
            return record.LastName + " " + record.FirstName + "(" + record.Code + ")";
        },
        fields: [
            {
                fieldName: "Id",
                fieldTitle: "شناسه",
                width: 100,
                defaultValueField: true
            },
            {
                fieldName: "FirstName",
                fieldTitle: "نام",
                width: 200,
                defaultDisplayField: true,
                filter: true,
                isStringType: true
            },
            {
                fieldName: "LastName",
                fieldTitle: "نام خانوادگی",
                filter: false,
                isStringType: true
            }
        ]
    });
</script>
}
```

شرح پارامترهای افزونه searchboxmvc.js

loadUrl: آدرس اکشن متدی است که بصورت ajax ای فراخوانی شده و نتایج حاصل را بازگشت میدهد. نتایج حاصله باید با فرمت json بازگشت داده شوند. اگر نتایج موفقیت باشد بصورت `Json(new { Status = "OK", })` بازگشت داده شوند و اگر خطایی در این بین صورت گرفت مقدار Status نباید مقدار OK باشد. پارامترهای مورد نیاز این اکشن نیز باید به ترتیب با نام های `fieldName` و `value` باشند که `fieldName` نام فیلدی است که جستجو بر اساس آن صورت می‌گیرد و `value` همان مقدار وارد شده توسط کاربر است. **defaultStringFilterMode**: اگر فیلد مورد جستجو از نوع رشته ای باشد (یعنی `isStringType` آن برابر true باشد) آنگاه پارامتر سوم اکشن متد بطور خودکار مقداردهی خواهد شد. مقادیر این خاصیت میتواند `startWith` یا `contains` یا `equal` باشد. **loadDataOnLeave**: اگر برابر false باشد، هر بار که متن input تغییر کرد بلافاصله یک تقاضا برای یافتن مقادیر به سرور فرستاده

میشود و نیازی نیست که فوکوس از کنترل خارج شود.

displayClass : نام کلاس css است که به 3 div اعمال خواهد شد.

displayNoResultClass : در صورتیکه جستجو نتیجه ای نداشته باشد این کلاس به 3 div اعمال خواهد شد.

display : یک فانکشن که برای ایجاد خروجی html برای نمایش در 3 div بکار می‌رود.

listItemsDisplay : یک فانکشن که برای ایجاد خروجی html برای آیتم‌ها بکار می‌رود.

fields : یک آرایه از فیلدهای موردنیاز پلاگین .

خاصیت‌های فیلد نیز بصورت زیر است:

fieldName : نام فیلد

fieldTitle : عنوان فیلد

defaultValueField : فیلد پیش فرض که جستجو بر اساس آن صورت می‌گیرد. اگر تعیین نشود فیلد اول آرایه به عنوان فیلد پیش فرض انتخاب خواهد شد.

defaultDisplayField : فیلد پیش فرض که برای نمایش متن 3 div بکار می‌رود(البته اگر پارامتر display تعیین نشود)

filter : اگر برابر true باشد این فیلد در لیست فیلدهای جستجو خواهد آمد و کاربر می‌تواند بر اساس آن جستجو انجام دهد.

isStringType : اگر برابر true باشد ، پارامتر سوم اکشن متد بطور خودکار مقداردهی خواهد شد.

لازم به ذکر است که این پلاگین کامل نیست و فقط برای ارائه مثال اینجا آورده شده است. هر یک از دوستان می‌توانند محتوای پلاگین را به سلیقه خود تغییر داده و پلاگین را کاملتر کنند.

[sample_mvc.zip](#)

نظرات خوانندگان

نویسنده: ح مراداف
تاریخ: ۱۶:۵۶ ۱۳۹۲/۱۱/۲۹

سلام ممنون از مطلب مفیدتون. چند تا سوال از خدمتتون داشتم: 1- آیا دیگه نیاز به قراردادن input نخواهیم داشت ؟ 2- فرض کنیم از این سیستم سرچ در بخشی از یک فرم استفاده کردیم و پس از کلیک روی دکمه submit نیاز داریم که به آی دی گزینه ای که توی بخش سرچ انتخاب شده ، دست پیدا کنیم ... چجوری باید به آی دی گزینه ای که توی کنترل سرچ انتخاب شده دست پیدا کنم ؟ با تشکر

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

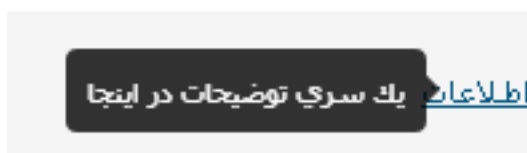
Tooltip یکی دیگر از کامپوننت‌های جاوا اسکریپتی Bootstrap است.

```
<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 تیب ندارد
        $(". " + this.settings.validClass).tooltip("destroy");
        // افزودن تول تیب 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، المان هایی 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 زمانیکه 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.

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: ۱۳۹۲/۰۴/۰۵ ۱۴:۴۱

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

+ [نحوه بازسازی وابستگی‌ها](#)

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 yesIsSelected، noIsSelected 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.

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 ۱۳:۰۰

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](#) »

نویسنده: حسین صفدری
تاریخ: ۱۳۹۳/۱۰/۲۳ ۱۱:۴۸

با سلام جسارتا جهت RTL منو بوت استرپ 3 به حالت چند سطحی باید کد بالا را به حالت زیر تغییر داد

```
.dropdown-submenu{position:relative}
.dropdown-submenu > .dropdown-menu{top:0;right:100%;margin-top:-6px;margin-right:-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:left;width:0;height:0;border-
color:transparent;border-style:solid;border-width:5px 5px 5px 0;border-right-color:#ccc;margin-
top:5px;margin-left:-10px}
.dropdown-submenu:hover > a:after{border-left-color:#fff}
```

```
.dropdown-submenu.pull-Right{float:none}
.dropdown-submenu.pull-Right > .dropdown-menu{right:-100%;margin-right:10px;-webkit-border-radius:6px 0 6px 6px;-moz-border-radius:6px 0 6px 6px;border-radius:6px 0 6px 6px}
```

[TwitterBootstrapMVC](#) یا به اختصار [BMVC](#) یک کتابخانه از Helper های مفید برای ساده سازی استفاده از Twitter Bootstrap در MVC می‌باشد .

در این کتابخانه امکانات مختلف Bootstrap از طریق Helper های نوشته شده برای MVC براحتی قابل استفاده می‌باشد و فرایند کد نویسی را ساده‌تر و در عین حال خواناتر می‌کند ، Helper های موجود در این کتابخانه به صورت زنجیره ای (fluent syntax) نوشته شده که استفاده از آن را سهولت می‌بخشد .

برای استفاده از آن در MVC 4 کافی است بعد از پیکر بندی Bootstrap ([راهنمایی](#)) به کتابخانه TwitterBootstrapMvc رفرنسی ایجاد کنید و با استفاده از این راهنما [نحوه استفاده](#) را فرا گیرید . همچنین می‌توانید آن را از طریق NuGet بارگذاری نمایید .

نسخه‌ی MVC4 آن را در اینجا برای شما نیز آپلود نمودم در زیر نمونه ای از استفاده از آن را می‌بینید

```
@Html.Bootstrap().LabelFor(x => x.UserName)
@Html.Bootstrap().TextBoxFor(m => m.UserName)
@Html.Bootstrap().PasswordFor(m => m.Password)
@Html.Bootstrap().FileFor(m => m.File)
@Html.Bootstrap().CheckBoxFor(m => m.IsActivated)
@Html.Bootstrap().RadioButtonFor(m => m.Gender, "male")
@Html.Bootstrap().DropDownListFor(m => m.State, Model.UsaStates)
@Html.Bootstrap().ListBoxFor(m => m.State, Model.UsaStates)
@Html.Bootstrap().TextAreaFor(m => m.Description)
```

ایجاد یک فرم

```
@using (Html.Bootstrap().Begin(new Form().Type(FormType.Inline)))
{
    @Html.Bootstrap().TextBoxFor(m => m.Email).Placeholder("Email")
    @Html.Bootstrap().PasswordFor(m => m.Password).Placeholder("Password")
    @Html.Bootstrap().CheckBoxFor(m => m.RememberMe).Label()
    @Html.Bootstrap().SubmitButton().Text("Sign in")
}
```

یک فرم Modal

```
@Html.Bootstrap().Button().Text("Show Modal").IconAppend(Icons.camera).TriggerModal("MyModal")

@using(var modal = Html.Bootstrap().Begin(new Modal() .Id("MyModal") .HtmlAttributes(new { @class
= "custom-class" })).Fade() )){
    using(modal.BeginHeader()){
        {
            <h2>Some header</h2>
        }
    }
    using(modal.BeginBody()){
        {
            <p>Some body<p>
        }
    }
    using(modal.BeginFooter()){
        {
            <p>Footer here.<p>
            @Html.Bootstrap().Button().Text("Close")
        }
    }
}
```


نظرات خوانندگان

نویسنده: Hamid NCH
تاریخ: ۱۳۹۲/۰۹/۱۳ ۱۲:۱۸

این کتابخانه محدودیت استفاده نداره؟ منظورم اینه کافیه این کتابخونه رو به رفرنس اد کنیم و استفاده کنیم! ممنون.

نویسنده: افشار محبی
تاریخ: ۱۳۹۲/۰۹/۱۳ ۱۲:۵۷

تا حالا فرصت نشده که به طور اساسی با Twitter Bootstrap درگیر بشوم. اما وجود این تعداد راهنما و ابزار جانبی آدم را امیدوار می کند که بتواند یک خروجی خیلی خوب از آن بگیرد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۳ ۱۲:۵۸

- داره ([مجاناً نیست](#)). البته فایل هایی که ایشون پیوست کردند به نظر محدودیت ندارند.
- ضمناً [باز هم هستند](#) یک سری Wrapper برای بوت استرپ که می توانند مورد استفاده قرار گیرند:
[TwitterBootstrapMvc](#) (نسخه سورس باز مطلب جاری است)
[Mvc Bootstrap Html Helper Extensions](#)
[Bootstrap Helpers](#) (معرفی در اینجا)
[Twitter Bootstrap Controls for ASP.NET](#)

نویسنده: رضا رضایی
تاریخ: ۱۳۹۲/۰۹/۱۳ ۲۳:۱۰

با سلام
متأسفانه این کتابخانه از نسخه 2 به بعد به صورت غیر رایگان عرضه شده است ، هر چند نسخه آپلود شده آخرین نسخه آن برای MVC 4 می باشد .
این کتابخانه برا اساس امکانات Bootstrap 3 نوشته شده است و هدفش تسهیل استفاده از Bootstrap 3 در MVC می باشد و بسیار کاربردیست .
دارای محدودیت خاصی نمی باشد و بسیاری از امکانات پیچیده Bootstrap را ساده نموده است

نویسنده: رضا رضایی
تاریخ: ۱۳۹۲/۰۹/۱۳ ۲۳:۱۴

twitter Bootstrap کتابخانه جامعی از CSS ها و JScript ها برای طراحی صفحات وب می باشد ک امروزه بسیار مقبول واقع شده است و ظاهری ساده و در عین حال کارآمد را برای وب سایت شما با پشتیبانی از انواع مختلف از نمایشگرها به ارمغان می آورد .

نویسنده: هومن
تاریخ: ۱۳۹۲/۰۹/۱۵ ۱۱:۲۸

سلام
مطلب بسیار مفیدی بود ممنون.
مشکلی که من دارم اینه که اولاً از طریق نیوگت قابل نصب نیست یعنی طبق سایت خودشون اسم بکجشو که مینویسم چیزی پیدا نمیکنه...
منم dll ها رو از سایت خودشون دانلود کردم و دستی add refrence کردم... در webconfig قسمت view ها هم دو تگ مربوط را اضافه کردم.

اما در ریزور تا اینجا **Bootstrap . Html** () را میشناسد اما بعد از پراتنز intelli-sense آن را نمیشناسد. مشکل از کجاست؟ ضمناً نسخه vs من 2013 و از mvc5 استفاده میکنم

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۵ ۱۱:۳۳

- نسخه سایت خودش سی روزه هست. باید یک فایل مجوز هم براش دریافت کنی، اینجا:
<https://www.twitterbootstrapmvc.com/Download> یا نسخه پیوست شده از متن رو دریافت و استفاده کن.
- برای MVC5 در اینجا: <http://www.nuget.org/packages/TwitterBootstrapMVC5>
- برای MVC4 در اینجا: <http://www.nuget.org/packages/TwitterBootstrapMVC>
[سایر موارد مرتبط](#)

نویسنده: رضا رضایی
تاریخ: ۱۳۹۲/۰۹/۱۶ ۱۹:۳۸

سلام ، از NuGet قابل دریافت ولی نیاز به لایسنس داره که باید بخری و نسخه مجانش فقط سی روزست ، فایل های که پیوست کردم این مشکل و نداره و آخرین نسخهست ، راه درستش اینه که اول Bootstrap 3 را کامل تو برنامهت پیکربندی کنی و از جواب دادنش اطمینان حاصل کنی بعد فقط کافیه فایل TwitterBootstrapMVC.dll و T4MVCExtensions.dll رفرنس بزنی ، بعد از اون باید در قسمت معرفی فضاهای نام در مسیر View / web.config این دو مورد و اضافه کنی

```
<add namespace="TwitterBootstrapMVC" />
<add namespace="TwitterBootstrap3" />
```

بعد از اون براحتی جواب میده
یادتون نره که فایل Portable.Licensing.dll هم هنگام اجرا باد در کنار دو فایل بالا وجود داشته باشد

نویسنده: مجتبی فخاری
تاریخ: ۱۳۹۲/۱۲/۲۳ ۱۷:۴۷

با سلام
آیا با Bootstrap.rtl3 هم جواب میده ؟
آخه من همین فایل شما را که برای mvc4 است دانلود کرده و add References هم نمودم و فضاهای نام را هم در View / web.config اضافه نمودم ولی اصلاً کار نمیده و برام Intelisense رو که میاره داخلش Bootstrap نداره.

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
```

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:

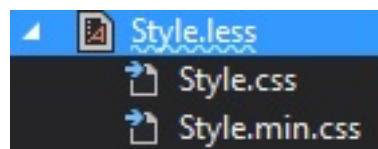
```
// Core variables and mixins
@import "variables.less";
@import "mixins.less";
.wrapper {
  .make-row();
}
.content-main {
```

```
.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: 8.333333333333333%;
  }
}
```

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 میکنن 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 = "{0} 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 پیش فرض با بوت استرپ 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](#)

من در یکی از پروژه‌ها از [Kendo UI Treeview](#) استفاده کردم و قصد داشتم قابلیت تغییر نام را به گره‌ها بدهم. به همین جهت پس از جستجو به [x-editable](#) برخوردم. این کتابخانه‌ی جاوااسکریپتی در ابتدا برای قالب‌های بوت استراپ طراحی شده بود که در حال حاضر اینگونه نیست و به راحتی در هر پروژه‌ای که فقط جی کوئری صدا زده شده باشد، قابل اجرا است و نسخه‌ی مخصوص [Angular](#) آن هم در [این آدرس](#) قرار دارد. همچنین این قابلیت اختیاری و پیش فرض را دارد که به طور خودکار اطلاعات تغییر یافته را به سمت url ایی که شما تعیین می‌کنید، ارسال کند. برای همین نیازی به استفاده جداگانه از jQuery Ajax برای ارسال اطلاعات نیست و البته در انتهای مقاله هم هنگام استفاده از درخت کندو به مشکلی برخورددم که آن را هم بررسی می‌کنیم.

وارد کردن کتابخانه‌ها

این کتابخانه شامل دو فایل css و JS می‌باشد که بسته به محیطی که در آن کار می‌کنید متفاوت هستند. در این صفحه شما می‌توانید برای 4 محیط Bootstrap2 , JQueryUI , JQuery و Bootstrap3 بسته‌ی مخصوصش را یا به صورت دانلود فایل‌ها یا از طریق [CDN](#) دریافت نمایید. در اینجا هم یک [دمو](#) از قابلیت‌های آن قابل مشاهده است.

برای شروع، کتابخانه‌ی مورد نظر خود را دریافت و آن‌ها را به صفحه‌ی خود اضافه نمایید. در صورتیکه از Bootstrap استفاده می‌کنید، ابتدا فایل‌های زیر را اضافه کنید:

```
<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet">
<script src="http://code.jquery.com/jquery-2.0.3.min.js"></script>
<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>
```

سپس فایل‌های x-editable را صدا بزنید.

اولین حالتیکه می‌توانید با این کتابخانه کار کنید، استفاده از خاصیت data-* است. نمونه زیر را در نظر بگیرید:

```
<a href="#" id="favsite" data-type="text" data-pk="1" data-url="@Url.Action(MVC.Categories.EditCategory())" data-title="Enter your favorite site">dotnettips.info</a>
```

به تعداد هر عنصری که نیاز است، اینکار را انجام دهید و به هر کدام یک id انتساب دهید. بعد از آن کد زیر را در قسمت script بنویسید:

```
$(document).ready(function () {
    $('#favsite').editable();
});
```

در صورتیکه قصد دارید خصوصیتی از اشیاء را که برای همه‌ی عنصرهای معرفی شده یکسان است، معرفی کنید می‌توانید از کد زیر بهره ببرید:

```
$(document).ready(function () {
    $.fn.editable.defaults.mode = 'inline';
    $('#favsite').editable();
});
```

کد بالا حالت ویرایش تمام عناصر معرفی شده را به inline تغییر می‌دهد.

حالت بعدی که می‌توان استفاده کرد به شکل زیر است:


```
<a href="#" id="favsite" >dotnettips.info</a>
```

```
$.fn.editable.defaults.mode = 'inline';
$(document).ready(function () {
    $('#favsite').editable({
        type: 'text',
        pk: 1,
        url: '@Url.Action(MVC.Categories.EditCategory())',
        title: 'Enter your favorite site'
    });
});
```

خوبی این روش این است که می‌توان اطلاعات بیشتری چون رویدادها را به آن پاس داد. تا الان با نحوه‌ی انتساب آن به اشیاء آشنا شدیم. اجازه دهید تا با خصوصیات آن آشنا شویم.

<p>همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید:</p> <pre>ajaxOptions: { type: 'put', dataType: 'json' }</pre>	AjaxOptions
<p>این ویژگی که تنها در حالت inline پاسخ می‌دهد، می‌تواند زمان بسته شدن x-editable را تغییر دهد که به طور پیش فرض با false مقداردهی شده است. جهت تغییر زمان بسته شدن، کد زیر را وارد نمایید:</p> <pre>anim: 'false' //or anim: { duration: 2000 }</pre>	Anim
در انتهای جدول آمده است.	autotext
<p>در صورتیکه عنصر مورد نظر محتوایی نداشته باشد و این خصوصیت را مقداردهی کنید، موقع ویرایش، این عبارت تعیین شده نمایش می‌یابد. در مثال بالا باید متن تگ a را حذف کرده تا نتیجه را ببینید: (البته فیلد value نباید مقداری داشته باشد)</p> <pre>defaultValue: 'default val' //or defaultValue: undefined //or defaultValue: null</pre> <p>در بقیه‌ی حالات، ویرایشگر خالی از متن خواهد بود و مقدار پیش فرض آن نال است.</p>	defaultValue
<p>false کردن این ویژگی باعث غیرفعال شدن x-editable بر</p>	disabled

<p>همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید:</p> <pre>ajaxOptions: { type: 'put', dataType: 'json' }</pre>	<p>AjaxOptions</p>
<p>روی کنترل جاری می‌گردد.</p>	
<p>خاصیت display یا مقدار بولین false را دریافت می‌کند، یا نال، یا یک تابع callback را می‌توان به آن پاس داد. این خصوصیت زمانی صدا زده می‌شود که اطلاعات به سمت آدرس سرور رفته و با موفقیت بازگشت داده می‌شوند (در صورتی که این ویژگی غیرفعال باشد، بلافاصله بعد از تایید کاربر، از اطلاعات وارد شده صدا زده می‌شود) و سپس متن جدید عنصر تغییر می‌یابد. حال اگر این خاصیت نال که مقدار پیش فرض آن است باشد، متن تغییر می‌یابد. ولی اگر false باشد، متن سابق باقی خواهد ماند و در صورتیکه تابعی به آن پاس داده باشید، طبق تابع شما عمل خواهد کرد.</p> <p>پارامترهایی که تابع شما می‌تواند داشته باشد به شرح زیر است:</p> <p>value : مقدار جدید</p> <p>response : پاسخ سرور (در صورتی که ارسال از طریق Ajax صورت گرفته باشد)</p> <p>و در صورتیکه عنصر شما select یا checlklist باشد که حاوی منبعی از مقادیر هست، مقادیرشان در قالب یک آرایه با نام sourceData بازگشت خواهد خورد. برای دسترسی به آیتم‌های انتخابی هم از کد زیر استفاده می‌کنیم:</p> <pre>\$.fn.editableutils.itemsByValue(value, sourceData)</pre>	<p>display</p>
<p>معرفی یک کلاس CSS برای موقعیکه عنصر خالی است.</p>	<p>emptyclass</p>
<p>در صورتی خالی بودن عنصر، این متن را برای عنصر نمایش بده.</p>	<p>emptytext</p>
<p>بعد از به روز رسانی متن عنصر، آن را با این رنگ highlight خواهد کرد و کد رنگی باید در مبنای هگز باشد. مقدار پیش فرض آن false است.</p>	<p>highlight</p>
<p>دو حالت نمایشی دارد که پیش فرض آن popup است و با باز کردن یک پنجره، مقدار جدید را دریافت می‌کند. مورد بعدی inline است که به جای باز کردن پنجره، متن عنصر را به حالت ویرایش تغییر می‌دهد.</p>	<p>mode</p>
<p>نام فیلدی که مقدارش تغییر می‌کند.</p>	<p>name</p>
<p>زمانی که کاربر فوکوس را از ویرایشگر می‌گیرد، ویرایشگر چه</p>	<p>onblur</p>

<p>همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید:</p> <pre>ajaxOptions: { type: 'put', dataType: 'json' }</pre>	<p>AjaxOptions</p>
<p>پاسخی باید به آن بدهد، باز بماند؟ <i>ignore</i> ، بسته شود؟ <i>cancel</i> و یا مقدار داده شده را تایید کند؟ <i>submit</i></p> <p>پارامترهای درخواست ایجکسی که کنترل در حالت پیش فرض ارسال می‌کند؛ شامل Pk که آن را با id رکورد پر می‌کنیم. name نام فیلدی که تغییر یافته است و value که مقدار جدید است. در صورتیکه دوست دارید اطلاعات اضافی‌تری نیز ارسال شوند، می‌توانید از این خاصیت استفاده کنید و پارامترها را در قالب Object به آن پاس کنید. ولی اگر بخواهید در کل همه‌ی پارامترها را رونویسی کنید باید یک تابع را به آن پاس کنید:</p> <pre>params: function(params) { // در این حالت پارامترهای پیش فرض ارسال نشده // و تنها پارامترهای معرفی شده در این تابع ارسال می‌شوند params.a = 1; return params; }</pre>	<p>params</p>
<p>کلید اصلی رکورد شما در دیتابیس یا هان id است. در صورتی که از کلیدهای ترکیبی استفاده می‌کنید، نگران نباشید فکر آن را هم کرده اند.</p> <pre>// کلید عدد pk:1, // کلید رشته ای pk:'dp123' // کلید ترکیبی pk:{id: 1, lang: 'en'}</pre> <p>معرفی یک تابع به آن و بازگشت</p> <pre>یکی از مقادیر بالا بعد از محاسبات pk:function() { }</pre>	<p>pk</p>
<p>این ویژگی فقط به درد حالت Popup می‌خورد که پنجره را کجای عنصر نمایش دهد و شامل چهار مقدار <i>left, right, top, bottom</i> می‌شود.</p>	<p>Placement</p>
<p>زمانی که مقدار جدید، برابر مقدار فعلی باشد و این خاصیت</p>	<p>saveonchange</p>

<p>همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید:</p> <pre>ajaxOptions: { type: 'put', dataType: 'json' }</pre>	<p>AjaxOptions</p>
<p>false باشد، هیچ تغییری رخ نخواهد داد. ولی اگر برابر true باشد، مقدار جدید اسال و جایگزین مقدار فعلی خواهد شد. مقدار پیش فرض آن false است.</p>	
<p>با استفاده از این خصوصیت در عنصر انتخابی به دنبال عناصری که در selector تعیین شده می‌گردد و حالت ویرایش را روی آن‌ها فعال می‌کند.</p> <p>در این حالت استفاده از خصوصیات emptytext و autotext در ابتدای امر ممکن نیست و بعد از اولین کلیک قابل استفاده هستند.</p> <p>نکته بعدی اینکه شما باید کلاس‌های زیر را دستی اضافه کنید. کلاس editable-click برای همه کنترل‌ها و کلاس editable-empty به کنترل‌های بدون مقدار و برای مقداردهی کنترل‌های بدون مقدار می‌توان از خاصیت data-value="" استفاده کرد.</p> <pre><div id="user"> <!-- empty --> Empty <!-- non-empty --> Operator </div> <script> \$('#user').editable({ selector: 'a', url: '/post', pk: 1 }); </script></pre>	<p>selector</p>
<p>سه مقدار auto,always و never را دریافت می‌کند. موقعی که شما آن را روی auto تنظیم کنید؛ در صورتی مقادیر به سمت سرور ارسال می‌شوند که دو خاصیت url و pk تعریف شده باشند. در غیر این صورت ویرایش فقط در حالت محلی و روی سیستم کاربر رخ خواهد داد.</p>	<p>send</p>
<p>در صورتیکه با false مقداردهی شود، تایید فرم به طور خودکار انجام می‌گیرد و اگر با یکی از مقادیر left یا Bottom پر شود، دکمه‌ها را در آن قسمت نشان می‌دهد.</p>	<p>showbuttons</p>
<p>اطلاعات به سمت سرور رفته و با موفقیت با کد 200 بازگشت داده شده‌اند. در مستندات نوشته است، هر کد وضعیتی غیر از</p>	<p>success</p>

<p>همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید:</p> <pre>ajaxOptions: { type: 'put', dataType: 'json' }</pre>	<p>AjaxOptions</p>
<p>200 بازگشت داده شود، به سمت خاصیت error هدایت می‌شود. ولی آن طور که من با httpresponsemessage تست کردم، چنین چیزی را مشاهده نکردم و مجدداً success صدا زده شد. پس بهتر هست داده‌ای را که به سمت کنترل برگشت می‌دهید، خودتان کنترل کنید. به خصوص اگر انتقال اطلاعات صحیح باشد. ولی اگر در دیتابیس، تغییر با خطا روبرو گردد بهتر است نتیجه‌ی آن ارسال شده و از تغییر مقدار فعلی ممانعت به عمل آورید.</p> <pre>success: function(response, newValue) { if(!response.success) return response.msg; }</pre>	
<p>اگر قصد دارید که باز و بسته کردن ویرایشگر را بر عهده‌ی مثلاً یک دکمه‌ی روی صفحه بگذارید، این خصوصیت به شما کمک می‌کند:</p> <pre>\$('#edit-button').click(function(e) { e.stopPropagation(); \$('#favsite').editable('toggle'); });</pre> <p>به جای toggle نیز می‌توان از show و hide هم استفاده کرد. وجود عبارت e.stopPropagation جهت باز شدن صحیح ویرایشگر الزامی است.</p>	<p>toggle</p>
<p>نوع ویرایشی را که قرار است انجام گیرد، مشخص می‌کند. text برای متن، date برای تاریخ، textarea جهت متون طولانی‌تر نسبت به text و بسیاری از موارد دیگر</p>	<p>type</p>
<p>این کلاس موقعی اعمال می‌گردد که اطلاعاتی را ویرایش کرده‌اید، ولی اطلاعاتی به سمت سرور ارسال نشده است. مثلاً pk مقداردهی نشده یا send=never قرار داید و یا اینکه به صورت محلی ذخیره می‌کنید و می‌خواهید در آخر همه‌ی اطلاعات را ارسال کنید.</p> <p>این خاصیت به طور پیش فرض با کلاس editable-unsaved مقداردهی شده که می‌توانید با نال کردن، از شرش خلاص شوید.</p>	<p>unsavedclass</p>
<p>این خاصیت با آدرس سمت سرور پر می‌شود. ولی می‌توان به آن یک تابع هم پاس کرد که این تابع جایگزین درخواست</p>	<p>url</p>

<p>همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید:</p> <pre>ajaxOptions: { type: 'put', dataType: 'json' }</pre>	<p>AjaxOptions</p>
<p>ایجکسی خودش خواهد شد و برای بازگشت دادن نتیجه‌ی این درخواست به سمت تابع‌های callback خودش می‌توانید یک deferred object را برگشت دهید:</p> <pre>url: function(params) { var d = new \$.Deferred; if(params.value === 'abc') { //returning error via deferred object return d.reject('error message'); } else { //async saving data in js model someModel.asyncSaveMethod({ success: function(){ d.resolve(); } }); return d.promise(); } }</pre>	
<p>مقدار پیش فرض آن نال است و می‌توان به آن یک تابع را جهت اعتبارسنجی سمت کلاینت پاس داد. به عنوان آرگومان، مقدار جدیدی را ارسال کرده و در آن به اعتبارسنجی می‌پردازید. در صورتی که مقدار، نامعتبر باشد، می‌توانید یک پیام خطا از نوع رشته‌ای را برگردانید.</p> <p>در صورتی که از نسخه‌ی 1.5.1 به بعد استفاده می‌کنید، دریافت یک object با مقادیر زیر نیز ممکن شده است:</p> <p>newValue: مقدار جدید و جایگزین مقدار غیر معتبر. msg: پیام خطا.</p> <p>به کدهای زیر در سه حالت نگاه کنید:</p> <pre>validate: function (value) { if (\$.trim(value) == '') { // تمامی نسخه‌های یک پیام متنی باز می‌گردد return 'This field is required'; } //1.5.1 // یک مقدار جدید برگشت میدهد که بلافاصله آن را تایید میکند و متن عنصر به روز می‌شود return { newValue: 'validated' }; } // متن جدید را ارسال کرده و پیام هشدار را نشان میدهد // ولی تایید نمی‌کند و منتظر تایید کاربر است return { newValue: 'validated',</pre>	<p>validate</p>

<p>همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید:</p> <pre>ajaxOptions: { type: 'put', dataType: 'json' }</pre>	<p>AjaxOptions</p>
<pre>msg: 'This field is required' }; }</pre>	
<p>مقدار پیش فرضی که در ویرایشگر نشان می‌دهد و اگر مقداردهی نشود، از متن عنصر استفاده می‌کند.</p>	<p>value</p>
<p>سه مقدار دارد auto (پیش فرض)، always و never. موقعیکه عنصر شما متنی نداشته باشد و روی auto تنظیم شده باشد، مقدار value را به عنوان متن عنصر نمایش می‌دهد. always کاری ندارد که عنصر شما متن دارد یا خالی است؛ مقدار value به آن انتساب داده خواهد شد. never هیچگاه.</p>	<p>autotext</p>

در قسمت بعدی که قسمت پایانی است مطالب را ادامه می‌دهیم.

در قسمت قبلی با نحوه‌ی اجرا و ویژگی‌های فنی و خصوصیات کدنویسی x-editable آشنا شدیم. غیر از این خصوصیات، خصوصیات دیگری هم هستند که فقط مختص نوع کنترلی هست که در قسمت type مشخص کرده‌اید.

کنترل‌های زیر جهت ورود اطلاعات در ویرایشگر پشتیبانی می‌شوند:

text
textarea
select
date
datetime
dateui
combodate
html5types
checkboxlist
wysihtml5
typeahead
typeaheadjs
select2
text

دکمه‌ای جهت حذف محتوای کادر متنی است. مقدار پیش فرض آن true است.	clear
برای دفاع در برابر کدهای مخرب html به کار می‌رود و کاراکترهای مدنظر را در صورت true بودن غیرفعال می‌کند. البته اگر از خاصیت display استفاده کنید این گزینه تأثیرش را از دست خواهد داد.	escape
یک کلاس css را به کادر متنی اعمال می‌کند.	inputclass
مقدار داده شده را در صورتی که کادر متنی خالی باشد، نشان می‌دهد.	placeholder
به معنی یک قالب. شما می‌توانید کد html تگ input خود را وارد کنید؛ ولی توصیه نمی‌شود.	tpl

TextArea

همان خاصیت‌های قبلی را دارد بعلاوه rows که نمایانگر مقدار ارتفاع آن است.

select

خاصیت‌های escape, input, class و tpl را دارد بعلاوه خاصیت‌های زیر:

همانند گزینه پایینی است ولی قبل از آن داده‌های خود را اضافه می‌کند.	prepend
از آنجا که یک لیست، لیستی از آیتم‌ها را دارد و کاربر یکی از آن‌ها را انتخاب می‌کند، این بخش، منبع آیتم‌ها را معرفی می‌کند. این خاصیت چهار نوع داده می‌پذیرد: آرایه یا شیءایی از مقادیر. تابعی که بعد از انجام هر عملی، اطلاعات به آن پاس می‌شوند و یا از نوع رشته که این رشته یک آدرس سمت سرور است که با درخواست از آن آدرس، اطلاعات را دریافت می‌کند.	source
اگر خاصیت بالا با آدرسی پر شده باشد که از سمت سرور بخواند، در دفعات بعدی مقدار دریافتی را از کش خواهد خواند.	sourceCache
یک پیام خطا هنگام بارگزاری اطلاعات	sourceError
در صورتیکه قصد اضافه کردن پارامتری را به درخواست ایجکسی دارید. یک شیء از پارامترها را به آن نسبت می‌دهیم و برای رونویسی پارامترها از یک تابع استفاده می‌کنیم که نحوه‌ی تغییرات را قبلاً در جدول شماره یک دیده‌اید.	sourceOptions

date

خاصیت‌های مشترک قبلی : `clear` و `tpl,input,class,escape` است.

پیکربندی تقویم را بر عهده دارد. برای اطلاعات بیشتر در مورد پیکربندی تقویم به این لینک مراجعه فرمایید.	datepicker
<code>{ weekStart: 0, startView: 0, minViewMode: 0, autoclose: false }</code>	
قالب بندی فرمت تاریخ جهت ارسال به سرور\ حالت پیش فرض <code>yyyy-mm-dd</code>	format
مقادیری که میتوان به کار برد: <code>yy yyyy mm m dd d</code>	
این فرمت هنگام نمایش به کار می‌آید و در صورتیکه مقدار عنصر در این قالب نباشد، آن را تبدیل می‌کند.	viewformat

datetime در بوت استراپ

کاملاً مشترک با مورد قبلی.

dateUI

مختص JQueryUI است و کاملاً مشترک با مورد قبلی.

combodate

موارد مشترک قبلی را دارد ولی به جای خاصیت `datepicker` از `combodate` استفاده می‌شود که پیکربندی آن در [این لینک](#) قرار دارد.

نوع‌های HTML 5

شامل موارد زیر است:

password
email
url
tel
number
range
time

[html5 شامل عناصر زیادی است](#) که ویژگی‌های جالبی را مد نظر دارند؛ ولی ممکن است بعضی المان‌ها در بعضی مرورگرها کارآیی مناسبی نداشته باشند که در [این صفحه](#) سازگاری مرورگرها با این نوع المان‌ها ذکر شده است. خاصیت‌های ذکر شده در مورد نوع text، در مورد آن‌ها نیز صدق می‌کند.

checkboxlist

همانند نوع select است؛ فقط خاصیت separator را دارد که کارش جدا کردن مقادیر است و مقدار پیش فرض آن علامت ' ' است.

wysihtml5

[سورس](#) و [دمو](#)ی این نوع ادیتور که بر پایه‌ی بوت استرپ بنا شده است و زحمت اضافه کردن کتابخانه‌ها به صفحه، بر عهده شماست.

مداخل زیر را به طور دستی به صفحه اضافه کنید:

```
<link href="js/inputs-ext/wysihtml5/bootstrap-wysihtml5-0.0.2/bootstrap-wysihtml5-0.0.2.css"
rel="stylesheet" type="text/css"></link>
<script src="js/inputs-ext/wysihtml5/bootstrap-wysihtml5-0.0.2/wysihtml5-0.3.0.min.js"></script>
<script src="js/inputs-ext/wysihtml5/bootstrap-wysihtml5-0.0.2/bootstrap-wysihtml5-
0.0.2.min.js"></script>
```

و همچنین اسکریپت x-editable برای کار با این عنصر را هم اضافه کنید:

```
<script src="js/inputs-ext/wysihtml5/wysihtml5.js"></script>
```

این فایل در بسته‌ای که دانلود کرده‌اید موجود است. شامل خاصیت‌های tp1, placeholder, inputclass, escape است و خاصیت wysihtml5 شامل تنظیمات و پیکربندی ادیتور است که پیکربندی آن را می‌توانید در [اینجا](#) مطالعه بفرمایید.

typeahead

این [گزینه](#) فقط مختص بوت استرپ 2 است و یک کنترل autocomplete به شمار می‌آید. منبع داده‌های آن از طریق خاصیت source به دو صورت آرایه و object تامین می‌گردد.

```
['text1', 'text2', 'text3' ...]
//or
[{value: 1, text: "text1"}, {value: 2, text: "text2"}, ...]
```

شامل خاصیت‌های tp1, inputclass, sourceCache, sourceError, sourceOptions, source, prepend, escape, clear است و شامل خاصیت typeahead جهت [پیکربندی](#) آن می‌شود.

typeaheadjs

همانند قبلی است و بر اساس [twitterBootstrap](#) است و شامل همان خصوصیات قبلی است. تنها خصوصیت typeahead آن است که باید از این [پیکربندی](#) استفاده کنید.

Select2

این المان بر اساس این [کتابخانه](#) سورس باز ایجاد می‌شود. و [مستندات آن](#) شامل جزئیات و پیکربندی آن می‌شود. برای معرفی آن فایل‌های زیر را به صفحه معرفی کنید.

```
<link href="select2/select2.css" rel="stylesheet" type="text/css"></link>
<script src="select2/select2.js"></script>
```

برای دریافت [استایل بوت استرپ](#) آن این فایل را صدا بزنید:

```
<link href="select2-bootstrap.css" rel="stylesheet" type="text/css"></link>
```

نکته: در حال حاضر خاصیت autotext روی این المان جواب نمی‌دهد و می‌توانید از خاصیت data-value به جای آن استفاده کنید.

شامل خاصیت‌های tpl , source , placeolder , escape , inputclass می‌باشد و از select2 برای [دریافت پیکربندی‌ها](#) ی کنترل استفاده می‌کند و علامت جدا کننده آن توسط viewseparator صورت می‌گیرد.

قابلی نو برای ویرایشگر

ویرایشگر فعلی ما به خصوص در بوت استرپ، ظاهر فوق العاده‌ای دارد. ولی اگر بازهم مایل به تغییر و رونویسی هستید، این امکان فراهم شده است.

\$fn.editableform.template

مقدار پیش فرض آن که حتما باید شامل تگ فرم و کلاس‌های مدنظر باشد:

```
<form>
  <div>
    <div><div></div><div></div></div>
    <div></div>
  </div>
</form>
```

در صورتی که قصد تغییر کلاس‌های آن را دارید باید کلاس‌های زیر را رونویسی کنید:

control-group

editable-input

editable-buttons

editable-error-block

\$fn.editableform.buttons

```
<button type="submit">ok</button>
<button type="button">cancel</button>
```

کلاس‌های editable-sumit و editable-cancel به طور خودکار به کلاس editable-buttons تزریق می‌شوند. و نهایتا جهت تغییر loading

\$fn.editableform.loading

<div></div>

گاهی اوقات نیاز است که خصوصیات این ویرایشگر را در شرایط متغیر صفحه کنترل کنیم، برای مثال گاهی پیش می‌آید که بخواهید در یک شرایط خاص ویرایشگر یک المان خاص را غیرفعال کنید. کد زیر مثال این تغییرات است.

```
$('#favsite').editable('option', 'disabled', false);
```

متدها و رویدادها

متدهایی که روی آن قابل اجراست:

ویرایشگر را بر اساس مقادیر اولیه روی عنصر مشخص شده فعال می‌کند.	editable
فوکوس را به input ویرایشگر باز می‌گرداند.	() activate
حذف ویژگی ویرایش از روی عنصر	() destory
غیرفعال کردن ویرایشگر	() disable
فعال سازی آن	() enable
<p>باعث بازگردانی مقدار جاری همه عناصر توسط شیء جفت کلید مقدار می‌شود و عناصری که شامل متن یا مقداری نیستند، از آن حذف می‌شوند. در صورتیکه قصد دارید مقدار تنها یک عنصر قابل دریافت باشد، با خاصیت isSingle آن را true کنید.</p> <pre>\$('#username, #fullname').editable('getValue'); //result: { username: "superuser", fullname: "John" } //isSingle = true \$('#username').editable('getValue', true); //result "superuser"</pre>	()getValue
مخفی کردن تگ فرم ویرایشگر	()hide
تغییر خصوصیات یک عنصر که در بالا هم نمونه کد آن را دیدیم.	(option(key,value
ست کردن مقدار جدید کنترل و پارامتر دوم وضعیت تبدیل این مقدار به فرمت داخلی است که برای آن تعریف شده است مثل date	(setvalue(value,convertStr
نمایش ویرایشگر	() show
<p>در صورتی که خاصیت ارسال خودکار به سمت سرور را غیر فعال کرده باشید، با این گزینه می‌توانید همه اطلاعات و تغییرات را ارسال کنید. برای ایجاد فرم بر اساس ویرایشگرها و ارسال اطلاعات با کلیک بر روی دکمه submit کاربرد دارد. یک مثال در این زمینه .</p> <p>پارامترهای options به شرح زیر هستند:</p> <pre>url data ajaxoptions</pre>	(submit(options

ویرایشگر را بر اساس مقادیر اولیه روی عنصر مشخص شده فعال می‌کند.	editable
error(obj) success(obj,config) از نسخه 1.5.1 میتوان این گزینه را به راحتی روی یک المان خاص هم صدا زد:	
<code>\$('#username').editable('submit')</code>	
کدی که صدا زده می‌شود بین دو وضعیت show و hide سوئیچ می‌کند.	() toggle
تغییر وضعیت بین دو حالت enable و disable	() toggleDisabled
انجام اعتبارسنجی بر روی همه کنترل ها. <pre> \$('#username, #fullname').editable('validate'); // possible result: { username: "username is required", fullname: "fullname should be minimum 3 letters length" } </pre>	() validate

رویدادها

این رویداد زمانی رخ می‌دهد که ویرایشگر دیگر قابل مشاهده نیست و شامل دو پارامتر event و reason است. reason دلیل اینکه چرا ویرایشگر از دید خارج شده است را با یکی از گزینه‌های زیر مشخص می‌کند.	
save cancel onblur nochange manual	hidden
<pre> \$('#username').on('hidden', function(e, reason) { if(reason === 'save' reason === 'cancel') { //auto-open next editable \$(this).closest('tr').next().find('.editable').editable('show'); } }); </pre>	
موقعی صدا زده میشود که متد editable روی عنصر صدا زده می‌شود و به یاد داشته باشید که این رویداد باید قبل از آن ست شده باشد.	
<pre> \$('#username').on('init', function(e, editable) { alert('initialized ' + editable.options.name); }); \$('#username').editable(); </pre>	init
موقعی که مقدار جدید، با موفقیت تایید می‌شود. دو پارامتر event و params را باز می‌گرداند که params شامل دو	save

<p>این رویداد زمانی رخ می‌دهد که ویرایشگر دیگر قابل مشاهده نیست و شامل دو پارامتر event و reason است. reason دلیل اینکه چرا ویرایشگر از دید خارج شده است را با یکی از گزینه‌های زیر مشخص می‌کند.</p> <p>save cancel onblur nochange manual</p>	hidden
<pre>\$('#username').on('hidden', function(e, reason) { if(reason === 'save' reason === 'cancel') { //auto-open next editable \$(this).closest('tr').next().find('.editable').editable('show'); } });</pre>	
<p>خصوصیت newValue و response است که به ترتیب مقدار جدید و اطلاعات برگشت داده شده از درخواست آژاکس است.</p> <pre>\$('#username').on('save', function(e, params) { alert('Saved value: ' + params.newValue); });</pre>	
<p>موقعیکه ویرایشگر نمایش می‌یابد و فرم با موفقیت رندر شده است. برای اشیایی چون select باید صبر کنید تا مقادیر آنها بارگذاری شوند.</p>	shown
<pre>\$('#username').on('shown', function(e, editable) { editable.input.\$input.val('overwriting value of input..'); });</pre>	

حل مشکل این ابزار در کندو

موقعیکه من این ابزار را بر روی treeview قرار دادم، به این مشکل برخورددم که اطراف پنجره باز شده، توسط حاشیه‌های treeview محدود شده است و مطابق شکل زیر قسمت‌هایی از آن دیده نمی‌شود. به همین علت css‌های کندو را به اندازه یک خط ویرایش کردم.



برای حل این مشکل فایل kendo.common-xxx را باز کنید. xxx بر اساس قالبی که برای کندو انتخاب کرده‌اید، می‌تواند متفاوت باشد. در مثال‌های کندو عموماً این xxx به نام default شناخته می‌شود یا برای مثال من، bootstrap بود. بعد از اینکه باز کردید، به دنبال چنین استایلی بگردید:

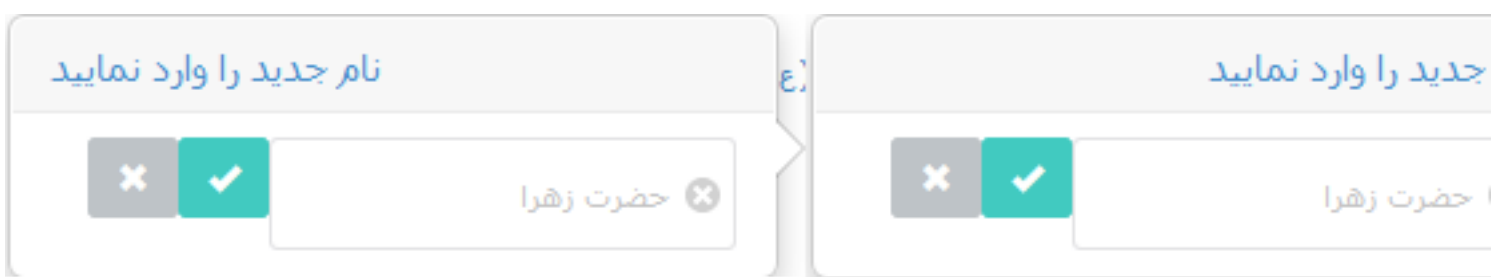
```
div.k-treeview{
border-width: 0px;
background: transparent none repeat scroll 0px center;
overflow: auto;
white-space: nowrap;
}
```

خط زیر را از آن حذف کنید تا مشکل حل شود.

```
overflow: auto;
```

نکته بعدی اینکه وقتی ویرایشگر در حالت popup قرار می‌گیرد، مقدار خاصیت title نمایش می‌یابد که عموماً با مضامینی چون "کلمه جدید را وارد نمایید" و ... پر می‌شود که به طور پیش فرض سمت چپ قرار گرفته است. کد زیر را در صفحه وارد کنید تا متن در سمت راست قرار بگیرد:

```
.popover-title {
text-align: right;
}
```



چند وقت پیش [لینکی](#) را معرفی کردم که در آن به طراحی پنجره های بوت استرپ 3 با استفاده از جی کوئری پرداخته بود و از آنجا که من دوست دارم انعطاف بیشتری در استفاده از این مدل کتابخانه ها داشته باشم و مستندات آن را حفظ نکنم، آن ها را به HtmlHelper تبدیل می کنم.

ابتدا از این آدرس فایل های مورد نظر را [دریافت کنید](#) . دو عدد از آن ها فایل استایل و دیگری فایل جی کوئری آن است که به ترتیب زیر صدا بزنید:

```
<script src="//code.jquery.com/jquery-1.11.3.min.js"></script>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>
<link href="~/content/css/bootstrap-dialog.min.css" rel="stylesheet"></link>
<script src="~/Scripts/bootstrap-dialog.min.js"></script>
```

[پروژه اصلی](#) شامل دو فایل اصلی است؛ یکی که در فضای نام models جهت قرار دادن مدل ها قرار گرفته و دیگری در فضای نام Controls جهت ایجاد متدهای Helper یا اجرایی قرار گرفته است.

ابتدا نیاز است که یک کلاس از نوع BootstrapDialog ایجاد کنید تا خصوصیات پنجره مشخص گردند. این خصوصیات به شرح زیر هستند:

```
var dialog=new BootstrapDialog();
dialog.Title="عنوان دیالوگ";
dialog.Message="متن پنجره";

//فعال سازی این خصوصیت باعث میشود یک دکمه بستن به
//پنجره اضافه شده و همچنین توسط کلیک کاربر در خارج از صفحه
//باعث بسته شدن پنجره شود یا استفاده از کلید
//ESC
dialog.Closable=false;

//تغییر اندازه دیالوگ
Dialog.Size=BootstrapDialogSize.SizeNormal;

//رنگ بندی دیالوگ را تغییر میدهد.مقدار زیر باعث میشود
//دیالوگ با رنگبندی قرمز نمایش داده شود تا برای نمایش خطاها مناسب باشد
Dialog.Type=BootstrapDialogType.Danger;

//برای اعمال کردن یک کلاس استایل دلخواه
Dialog.CssClass="";

//آیکن برای دیالوگ-استفاده از نام کلاس آیکن های بوت استرپ
Dialog.SpinnerIcon="";

//یک توصیف است که فقط در کد صفحه نمایش داده میشود
//استفاده خاصی ندارد
Dialog.Description="";

//بعد از بستن دیالوگ ، کدهای آن در صفحه حذف خواهند شد
//اگر میخواهید کد را بارها و بارها نمایش دهید
//آن را با مقدار ناصحیح مقدار دهی کنید
dialog.AutoDestory=false;

//===== رویدادها =====
//این رویداد قبل از نمایش دیالوگ نمایش داده میشود
```

```
dialog.OnShow="function(){alert('before Dialog');}";

// این رویداد بعد از نمایش دیالوگ اجرا می‌شود
dialog.OnShown="function(){alert('after Dialog shown');}";

// موقع درخواست بستن دیالوگ قبل از بسته شدن اجرا می‌گردد
dialog.OnHide="function(){alert('before Dialog close');}";

// بعد از بسته شدن دیالوگ اجرا می‌شود
dialog.OnHidden="function(){alert('after Dialog close');}";
```

تا اینجا خصوصیات پنجره معرفی شده است. در حال حاضر نیاز است که کدهای آن در قسمت View درج شوند برای اینکار از یک Helper کمک می‌گیریم:

```
@{
var dialog=new BootstrapDialog();
dialog....
// .....
}
@HTML.BootstrapDialog("example1",dialog)
```

در کد، اولین پارامتر نام پنجره است: از این اسم بعداً می‌توانید جهت اجرای متدها، چه دستی توسط خود شما یا ایجاد متدهای ساده توسط خود کلاس استفاده کنید. دومین پارامتر هم دریافت خصوصیات پنجره است که در بالا توضیح دادیم.

دکمه ها

در صورتیکه قصد دارید دکمه‌ای را روی پنجره ایجاد نمایید، با شیوه زیر اینکار صورت می‌گیرد:

```
var dialog=new BootstrapDialog();
var cancelButton=new BootstrapDialogButton("cancelButton");
//cancelButton.id="cancelButton";
cancelButton.label="Cancel";
cancelButton.Key=65;
cancelButton.Action="function(){alert('You Clicked!');}";
dialog.AddButton(cancelButton);
```

برای حذف آن هم می‌توانید به صورت زیر اقدام کنید:

```
dialog.RemoveButton("cancelButton");
```

داده ها

در صورتیکه قصد دارید داده‌هایی را به این پنجره نسبت دهید تا بعداً در کدهای سمت کلاینت از آن استفاده کنید می‌توانید از کد زیر استفاده کنید:

```
dialog.AddData("key","value");
```

جهت حذف:

```
dialog.RemoveData("key");
```

متدها

متدها را به دو صورت می‌توانید اعمال کنید:

دستی: که می‌توانید اطلاعات متدها را در همان صفحه [مثال و مستندات](#) ببینید و از نامی که به دکمه‌ها و پنجره‌ها می‌دهید آن‌ها را

اعمال کنید.

با استفاده از کلاس: کلاس ما شامل دو متد دیگر برای کنترل متدها می باشد. حدود 13 متد در آن پشتیبانی می شود که باعث می شود در بسیاری از اوقات دیگری نیازی به دانستن نام متدها نداشته باشید. یکی از متدها برای استفاده در Helper طراحی شده است که خروجی آن از نوع MvcHtmlString است و متد دیگر خروجی string دارند که می توانید در صورتیکه خواستید، در رویدادها و خارج از Html Helper از آن استفاده کنید.

نحوه ی استفاده از helper به شکل زیر است؛ فرض شده است که پنجره را تشکیل داده اید و الان قصد دارید با کلیک بر روی یک دکمه آن را نمایش دهید:

```
$( "#btnshowpopup" ).click(function() {
    @HTML.RunBootstrapDialogMethod("example1",BootstrapDialogMethods.Open)
});
```

البته بعضی از متدها شامل ورودی یا آرگومان هستند که در کامنت مربوط به آن، تعداد پارامترها و ترتیب آن ها ذکر شده است. یک نمونه از آن:

```
$( "#btnshowpopup" ).click(function() {
    @HTML.RunBootstrapDialogMethod("example1",BootstrapDialogMethods.SetData,new{"key","value"})
});
```

برای استفاده از متد دوم که خروجی آن از نوع string است، می توان از آن در بین کد رویدادها استفاده کرد. مثال زیر ایجاد یک رویداد، برای یکی از دکمه های پنجره است که با کلیک بر روی آن، پنجره بسته می شود:

```
cancelButton.Action="function(){{{0}}}";
cancelButton.Action=string.format(cancelButton.Action,RunBootstrapDialogMethod("example1",BootstrapDialogMethods.Close));
```

به عنوان یک مثال نهایی کد زیر را نوشته که نتیجه آن را در تصویر زیر می بینم:

```
@{
    const string dialogName = "errorDialog";
    var cancelButton = new BootstrapDialogButton();
    cancelButton.Id = "btncancel";
    cancelButton.Label = "بستن";
    cancelButton.Action = "function(){{{0}}}";
    cancelButton.Action = String.Format(cancelButton.Action,
    Dialogs.RunBootstrapDialogMethod(dialogName, BootstrapDialogMethods.Close));

    var dialog = new BootstrapDialog();
    dialog.AddButton(cancelButton);
    dialog.Title = "عنوان";
    dialog.Message = "پیام هشدار";
    dialog.DialogType=BootstrapDialogType.Warning;
    dialog.DialogSize=BootstrapDialogSize.SizeNormal;
    dialog.Closable = false;
    dialog.AddData("data1","5");
}

@Html.BootstrapDialog(dialogName, dialog)
@Html.RunBootstrapDialogMethod(dialogName,BootstrapDialogMethods.Open);
```

عنوان ×

پیام هشدار

بستن

نکته مهم: برای ایجاد پنجره از طریق توابع عمل کنید و خط تعریف پنجره را داخل یک تابع قرار داده و از همانجا آن را باز کنید. در حال حاضر به نظر میرسد در صورتی که تعریف پنجره به طور عمومی باشد، این کتابخانه برای بار دوم به بعد مشکلاتی دارد که مشکل آن بسته نشدن پنجره است. در حال حاضر در گیت هاب این مسئله را عنوان کردیم، در صورتی که پاسخی ارائه شود همینجا به اطلاع شما می‌رسانم.

عنوان:	ایجاد BootstrapSwitch در MVC
نویسنده:	علی یگانه مقدم
تاریخ:	۲۰:۱۵ ۱۳۹۴/۰۴/۱۹
آدرس:	www.dotnettips.info
گروه‌ها:	MVC, jQuery, Twitter Bootstrap

در [مقاله‌ی قبلی](#) ما بخشی از BootstrapDialog را با استفاده از [Reflection](#) پیاده سازی کردیم. دلیل اینکه پیاده سازی کاملی از آن نداشتیم، متغیر بودن مقادیر و پیچیده‌تر شدن و طولانی تر شدن کد نویسی آن بود که برای آن کد ارزش زیادی نداشت تا وقت بیشتری صرف شود. ولی در اینجا بخاطر پیچیدگی کمتر، به طور کامل از Reflection استفاده شده است.

شیء [BootstrapSwitch](#) یک چک باکس است که با استفاده از جی کوئری و استایل‌ها به یک سوئیچ انیمیشنی زیبا تبدیل شده است که خودم به شخصه علاقه زیادی به استفاده‌ی از آن در پروژه‌های شخصی پیدا کرده‌ام. غیر از زیبایی، حس خوبی از کارکرد برنامه میدهد. [فایل‌های موردنیاز](#) را دانلود کرده و آن‌ها را در ابتدای صفحه و با رعایت ترتیب صدا بزنید:

```
<script src="//code.jquery.com/jquery-1.11.3.min.js"></script>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>
<link href="~/content/css/bootstrap-switch.min.css" rel="stylesheet"></link>
<script src="~/Scripts/bootstrap-switch.min.js"></script>
```

نکته مهم: فایل css شامل دو نسخه هست که یکی از آن برای Bootstrap2 و دیگری برای نسخه 3 آن است که نسبت به آن نسخه، استایل مناسب را انتخاب کنید. [پروژه‌ی اصلی](#) را دریافت کنید و آن را به solution خود اضافه کنید. پروژه به دو بخش اصلی Models و Controls تقسیم می‌شود که بخش مدل آن، برای ایجاد ساختارهای آن و در بخش کنترل، برای ترسیم آن به صورت HtmlHelper به کار می‌رود.

ابتدا قبل از هر چیزی یک شیء از کلاس BootstrapSwitchModel ایجاد کنید و مقادیر دلخواه خود را به خصوصیت‌های آن نسبت دهید:

```
var model=BootstrapSwitchModel();

//وضعیت فعال بودن و غیرفعال بودن سوئیچ
model.Checked=true;

//اندازه آن
model.Size=BootstrapSize..normal;

//یک انیمیشن ساده موقع سوئیچ کردن دارد
model.Animate=true;

//به چک باکس عادی تبدیل میشود
model.Disabled=true;

//غیرفعال شده و به صورت فقط خواندنی قابل دسترس است
model.Readonly=true;

//رنگ فعال بودن
model.OnColor=BootstrapColor.Success;

//رنگ غیرفعال بودن
model.OffColor=BootstrapColor.Danger;

//متن نمایشی در هنگام فعال بودن
model.OnText="On";

//متن نمایشی در حالت عدم انتخاب
model.OffText="Off";

//بین دو حالت روشن و خاموش نمایش داده میشود
model.Label="Public Display";

//تعیین میزان اندازه برجسب بالا
model.LabelWidth=100;

//سوئیچ به صورت آینه ای معکوس میشود
model.Inverse=false;

//کلاسی جهت تغییر استایل سوئیچ
model.BaseClass="myclass";

//تعیین کلاس برای تگ اصلی پدر
model WrapperClass="wclass";
```

```
// فقط یکی از چند سوئیچ میتواند فعال باشد  
model.RadioAllOff=false;  
  
// یک سوئیچ در حالت عادی فقط یکی از  
// وضعیت‌ها را نمایش میدهد ولی در این حالت  
// سوئیچ در ابتدا بین این دو وضعیت گیر کرده است  
model.Indeterminate=true;  
  
// اندازه سمت چپ و راست سوئیچ  
model.HandleWidth=25;
```

برای ترسیم آن در یک ویو هم به صورت زیر عمل کنید:

```
@{  
    var model=BootstrapSwitchModel();  
    ....}  
  
@HTML.BootstrapSwitch("id",model);
```

برای اطلاع از رویدادهای این کنترل، [مستندات آن](#) را مطالعه کنید و از id برای ارتباط با آن استفاده می‌کنند.

عنوان:	Gulp #2
نویسنده:	م محمد شریفی
تاریخ:	۱۰:۴۵ ۱۳۹۴/۰۶/۱۷
آدرس:	www.dotnettips.info
گروه‌ها:	JavaScript, Twitter Bootstrap, bower, Gulp

در [قسمت قبلی](#) بحث کردیم که گالپ چیست و چه کاربردی دارد و در نهایت آن را بر روی سیستم خود نصب کردیم. در این مقاله و مقالات بعد می‌خواهیم کار خود را با راه اندازی یک workflow برای بوت استرپ، روند شخصی سازی آن را بسیار آسان و لذت بخش‌تر کنیم. امیدوارم که برای ادامه‌ی این بحث هیجان انگیز آماده باشید!

ساخت پروژه گالپ

ابتدا یک پوشه‌ی دلخواه به نام **project** را درست کنید. سپس خط فرمان خود را به این مسیر تغییر دهید و در نهایت دستور زیر را وارد کنید:

```
npm init
```

این دستور برایمان یک فایل *package.json* می‌سازد تا هم مشخصات پروژه را مثل نام، ورژن، نام توسعه دهنده، مخزن و ... مشخص کنیم و هم وابستگی‌های آن را، تا توسعه دهندگان دیگر، هنگام استفاده از پروژه، با مشکل مواجه نشوند و فقط با اجرای دستور `npm install` تمام وابستگی‌های پروژه را نصب کنند. حتما مشاهده کرده‌اید که این دستور چند سوالی را از شما می‌پرسد. برای نمونه من آنها به این صورت پاسخ می‌دهم و در نهایت از من یک تایید می‌گیرد که **yes** را می‌زنم.

```
name: (Gulp-RTLbootstrap-fontawesome)
version: (1.0.0)
description: An Awesome workflow
entry point: (index.js) index.html
test command: test
git repository: https://github.com/mmdsharifi/gulp-rtlBootstrap-fontawesome.git
keywords: gulp, rtlbootstrap, persian bootstrap
author: Mohammad Sharifi
license: (ISC) MIT
```

الان فایل *package.json* درست شده و چون ما در این پروژه می‌خواهیم از گالپ استفاده کنیم، یکی از وابستگی‌های پروژه‌ی ما گالپ خواهد بود. بدین معنی که اگر بخواهیم پروژه را توسعه دهیم و گالپ نصب نشده باشد، با مشکل مواجه می‌شویم.

نصب گالپ

در خط فرمان دستور زیر را وارد کنید تا گالپ در این پروژه نصب شود.

```
npm install gulp --save-dev
```

تفاوتی آن با دستوری که در مقاله‌ی قبلی اجرا کردیم، این است که این دستور فقط گالپ را در مسیر جاری در فولدر *node_modules* نصب می‌کند و **save-dev** آن را به وابستگی‌های توسعه‌ی پروژه در فایل *package.json* اضافه می‌کند. گام بعدی، ساخت فایلی است که گالپ به آن نیاز دارد تا با استفاده از آن، تسک‌ها و کارهایی را که برایش نوشته‌ایم، از آن بخواند و اجرا کند.

ایجاد فایل *gulpfile.js*

این فایل را یا به صورت دستی ایجاد کنید یا با خط فرمان با دستور ذیل:

```
touch gulpfile.js
```

و حالا آن را در ویرایشگر مورد علاقه‌ی خود باز کنید. من از ویرایشگر [Atom](#) استفاده می‌کنم. اما notepad هم کفایت می‌کند.

نوشتن اولین تسک گالپ

در ویراشگر خط زیر را می‌نویسیم:

```
var gulp = require('gulp');
```

require به Node می‌گوید که به فولدر node_modules برای پکیج gulp نگاه کند. زمانیکه آن را پیدا کرد، آن را به متغیر gulp انتساب می‌دهیم تا از تابع‌های گالپ بتوانیم استفاده کنیم. حال می‌خواهیم اولین تسک خود را بنویسیم:

```
gulp.task('task-name', function() {
  // Stuff here
});
```

گالپ دارای [۴ تابع](#) task, src, dest, watch است که با آنها آشنا خواهیم شد. task یک کار را برای گالپ تعریف می‌کند و [سه پارامتر](#) دارد. اولی نام تسک، دومی (اختیاری) وابستگی این تسک (بدین معنا که اول باید این وظیفه اجرا شود، سپس تسک جاری) و در نهایت تابع درون تسک‌ها را می‌نویسیم. برای مثال:

```
gulp.task('hello', function() {
  console.log('Hello Gulp !');
});
```

فایل را ذخیره کنید. می‌خواهیم به گالپ بگوییم که این وظیفه را انجام دهد. کافی است **gulp hello** را در خط فرمان وارد کنیم. نتیجه به صورت زیر خواهد بود:

```
mmdsharifi@mmdsharifi ~/Work/Gulp-RTLbootstrap-fontawesome $ gulp hello
[09:10:44] Using gulpfile ~/Work/Gulp-RTLbootstrap-fontawesome/gulpfile.js
[09:10:44] Starting 'hello'...
hello Gulp !
[09:10:44] Finished 'hello' after 83 μs
```

البته که تسک‌هایی که برای گالپ می‌نویسیم، کاراتر از این است؛ برای مثال:

```
gulp.task('task-name', function () {
  return gulp.src('source-files') // Get source files with gulp.src
    .pipe(aGulpPlugin()) // Sends it through a gulp plugin
    .pipe(gulp.dest('destination')) // Outputs the file in the destination folder
});
```

با استفاده از متد src به گالپ می‌گوییم که مسیر مبدأ فایل‌ها، برای انجام تسک کجا است و dest هم بعد از انجام تسک، فایل‌های خروجی را به مقصد مشخص می‌برد. متد pipe یک تابع ند جی اس است که مطابق [مستندات](#) خودش متدی است که تمام [جریان](#)‌های قابل خواندن را واکنشی می‌کند و به مسیری [که به صورت آرگومان به عنوان مقصد] داده شده، هدایت می‌کند. شاید در ابتدا نوشتن تسک برایتان کمی پیچیده باشد، اما بعد از ساخت اولین پروژه با گالپ، خواهید دانست که تسک نوشتن برای گالپ کاری بسیار آسان و شیرین است!

[مخزن پروژه](https://github.com/mmdsharifi/gulp-rtlBootstrap-fontawesome) در گیت هاب : <https://github.com/mmdsharifi/gulp-rtlBootstrap-fontawesome>

نام کامیت این قسمت: Init commit

در مقاله بعدی gulp را در کنار [bower](#) بکار خواهیم برد. بهتر است مطالعه‌ای در مورد bower نیز انجام دهید. (پیشنهاد: [+ و +](#))

نظرات خوانندگان

نویسنده: محمود راستین
تاریخ: ۱۹:۹ ۱۳۹۴/۰۷/۰۲

تشکر بابت این مقاله. فقط یک نکته رو بیان کنم که فرمان برای پلتفرم ویندوز نیست. اگر دستور :

```
touch gulpfile.js
```

رو صادر کنیم با خطای زیر رو به رو میشیم :

```
'touch' is not recognized as an internal or external command, operable program or batch file.
```

برای ایجاد فایل خالی در ویندوز باید اینکارو بکنیم :

```
echo $null >> gulpfile.js
```

این دستور همانند دستور touch هستش. با این دستور فایل gulpfile.js در مسیر پروژه ساخته میشه.

نویسنده: م محمد شریفی
تاریخ: ۹:۲۳ ۱۳۹۴/۰۷/۰۳

خواهش می‌کنم.

بله درسته دستور touch در لینوکس به درستی کار می‌کند.

سپاس از اشتراک گذاری این نکته.

در [قسمت اول](#) گالپ را معرفی کردیم و در [مقاله قبلی](#) به نوشتن اولین تسک با گالپ پرداختیم. در این قسمت می‌خواهیم با نصب bower، پروژه‌ی workflow بوت استرپ راستچین شده را انجام دهیم.

نصب bower



[bower](#) یک مدیریت پکیج سمت Front end است و از مزایای استفاده از آن می‌توان به موارد زیر اشاره کرد:

ساده کردن تعریف وابستگی‌های منابع پروژه با تعریف یک فایل **bower.json**

نیازی به commit کردن وابستگی‌های پروژه نیست.

با ذکر ورژن مربوط به وابستگی یا محدوده‌ی قابل قبول برای آن، به روز رسانی منابع به سادگی با یک دستور انجام می‌شود.

وابستگی‌های وابسته به یک منبع را نیز نصب می‌کند. برای مثال زمانیکه بوت استرپ را به عنوان وابستگی پروژه تعریف می‌کنیم،

وابستگی آن یعنی jquery را چون در فایل **bower.json** بوت استرپ تعریف شده‌است، به صورت خودکار دانلود می‌کند.

در نهایت افراد هم تیمی یا توسعه دهندگان دیگر به راحتی با زدن دستور **bower install** تمام وابستگی‌های پروژه را می‌توانند نصب کنند.

برای نصب آن کافی است دستور زیر را بزنید و بعد از نصب نیز دستور خط دوم را در مسیر پروژه وارد کنید تا یک فایل

bower.json را برایمان بسازد. برای اینکار به سوال‌هایی که می‌پرسد باید جواب دهیم. تنها نکته‌ای که قابل ذکر است، پاسخ به

سوال **what types of modules does this package expose ?** است که باید گزینه‌ی Node را انتخاب کنید.

```
sudo npm install -g bower
bower init
```

حال می‌خواهیم وابستگی‌های پروژه را نصب کنیم که عبارتند از **bootstrap-sass,fontawesome,bootstrap-rtl** :

```
bower install bootstrap-sass-official --save
bower install fontawesome --save
bower install bootstrap-rtl --save
```

نکته : عبارت `--save` وابستگی مربوطه را به `bower.json` اضافه می‌کند. اگر نصب با موفقیت صورت گرفته باشد، پکیج‌های مربوطه را می‌توانید در فولدر `bower_components` در `root` پروژه مشاهده کنید.

نصب پلاگین‌های مورد نیاز gulp

ما می‌خواهیم که بوت استرپ و نگارش `sass` آن‌را کامپایل کنیم و همچنین وابستگی‌های `bower` پروژه را از طریق گالپ نصب کنیم تا نیازی به زدن `bower install` نباشد و توسعه دهنده‌ی پروژه فقط با زدن `npm install`، تمام وابستگی‌های پروژه‌ی ما را نصب کند. می‌توان تمام پلاگین‌ها را پشت سر هم با یک دستور نصب کرد و یا به صورت جداگانه این کار را انجام داد.

```
sudo npm install gulp gulp-ruby-sass gulp-notify gulp-bower --save-dev
```

نکته: پلاگین `gulp-notify` به منظور نشان دادن خطاها در ترمینال است؛ تا در صورت وجود اشتباه در کامپایل فایل‌های `Sass`، کل روند گالپ متوقف نشود.

نکته ۲: برای اینکه کامپایل `sass` انجام شود نیاز به `Ruby` دارید. برای ویندوز می‌توانید از [روبی اینستالر](#) استفاده کنید.

نوشتن تسک‌ها برای گالپ

به قسمت مهم و هیجان انگیز کار رسیدیم! همان طور در مقاله قبلی گفتیم، ابتدا باید ماژول‌هایی را که نصب کردیم، `include` کنیم به این صورت:

```
var gulp = require('gulp'),
    sass = require('gulp-ruby-sass'),
    notify = require('gulp-notify'),
    bower = require('gulp-bower');
```

برای اینکه دسترسی به مسیرهای مهم پروژه آسان‌تر شود، آن‌ها را درون یک شیء نگه داری می‌کنیم.

```
var config = {
  sassPath: './resources/sass',
  bowerDir: './bower_components'
}
```

تسک `baor` را اضافه می‌کنیم تا کار `bower install` را خودکار کنیم. مزیت این کار این است اگر یک هم تیمی، پکیج جدیدی را در حین توسعه‌ی پروژه نصب کرد، بدون اینکه لازم باشد تا در جایی از پروژه، بقیه را از آن مطلع کنید، فقط با زدن `gulp` خیالتان راحت شود که تمام کارهایی که باید انجام دهید، گالپ برایتان انجام می‌دهد.

```
// create a task to do bower install
gulp.task('bower', function() {
  return bower()
    .pipe(gulp.dest(config.bowerDir))
});
```

در گام بعدی، تسک `جاوا اسکریپت` را اضافه می‌کنیم. یعنی جی کوری و فایل `bootstrap.js` را به مسیر `public/js` می‌آوریم. فولدر `public` برای جدا سازی فایل‌های نهایی از فایل‌های توسعه است و به همین صورت برای فونت آیکن‌های `fontawesome`.



```
// Copy js files to public folder
gulp.task('js', function() {
  return gulp.src([config.bowerDir + '/bootstrap-sass-official/assets/javascripts/bootstrap.min.js',
    config.bowerDir + '/jquery/dist/jquery.min.js'
  ])
  .pipe(gulp.dest('./public/js'));
});

// Copy fontawesome icons to public/fonts folder
gulp.task('icons', function() {
  return gulp.src(config.bowerDir + '/fontawesome/fonts/**/*.*)
  .pipe(gulp.dest('./public/fonts'));
});
```

برای سی اس اس هم تسک مربوطه اش را به صورت زیر می نویسیم



```
gulp.task('css', function() {
  return sass(config.sassPath + '/style.scss', { // Our custom sass
    style: 'compressed', // minify css
    loadPath: [ // load paths to easily use imports in resources/sass
      './resources/sass',
      config.bowerDir + '/bootstrap-sass-official/assets/stylesheets', // bootstrap sass files
      config.bowerDir + '/fontawesome/scss' // awesome icons sass files
    ]
  });
});
```

حال تعریف می‌کنیم که اگر خطایی در حین کامپایل رخ داد، آن را به ما نشان دهد و در نهایت فایل کامپایل و فشرده شده را در مسیر خروجی قرار می‌دهیم. کدها را به صورت زیر به روز می‌کنیم

```
gulp.task('css', function() {
  return sass(config.sassPath + '/style.scss', { // Our custom sass
    style: 'compressed', // minify css
    loadPath: [ // load paths to easily use imports in resources/sass
      './resources/sass',
      config.bowerDir + '/bootstrap-sass-official/assets/stylesheets', // bootstrap sass files
      config.bowerDir + '/fontawesome/scss' // awesome icons sass files
    ]
  })
  .on('error', notify.onError(function(error) {
    return 'Error: ' + error.message;
  }))
  .pipe(gulp.dest('./public/css'));
});
```

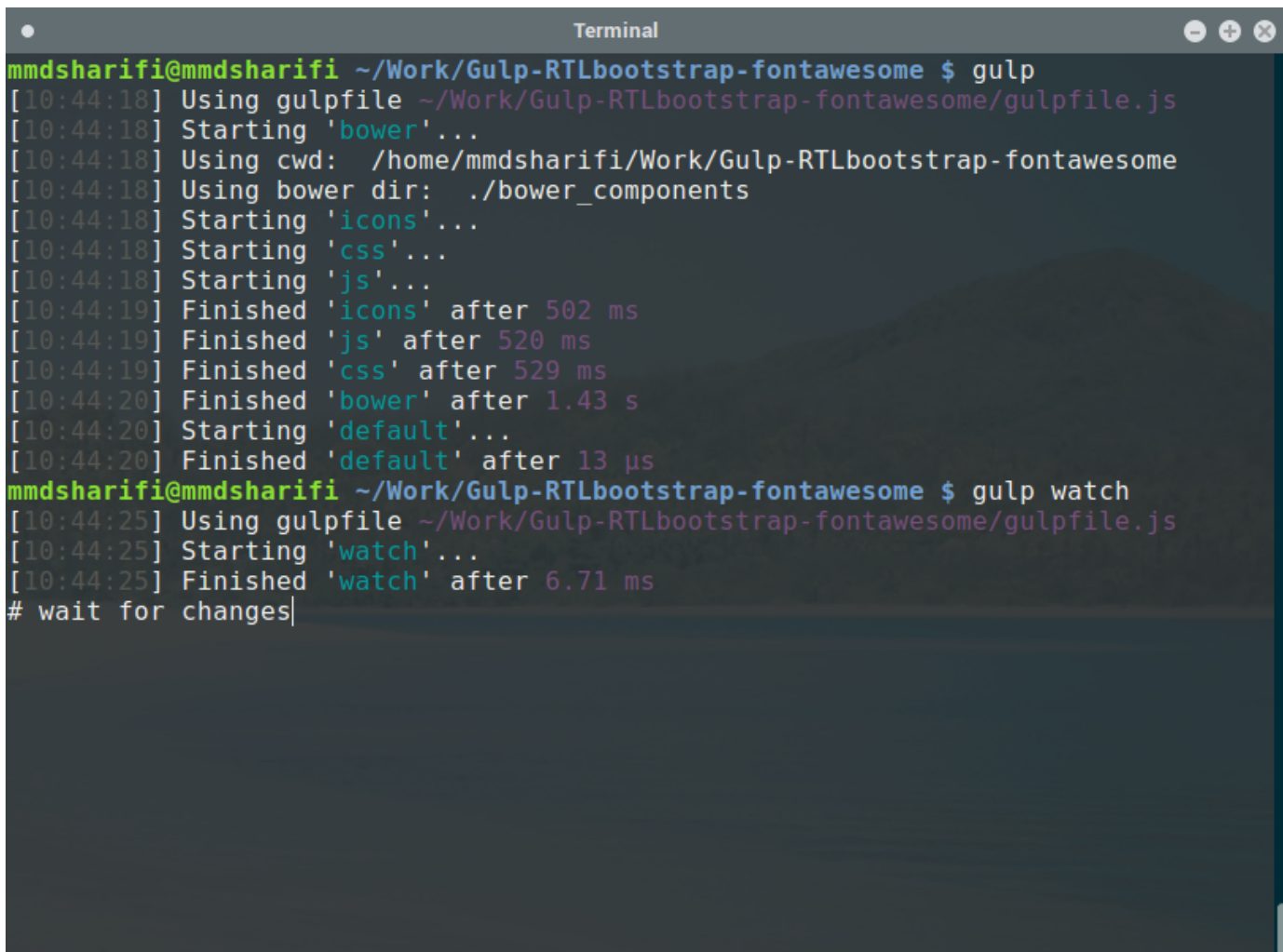
اینجا ما می‌خواهیم که کارها را خودکار سازی کنیم تا با تغییر و ذخیره‌ی مجدد فایل‌های سس، تسک سی اس اس را انجام دهد. برای این کار کدهای زیر را اضافه می‌کنیم

```
// Rerun the task when a file changes
gulp.task('watch', function() {
```

```
gulp.watch(config.sassPath + '/*.scss', ['css']);
});
```

در نهایت برای سادگی می‌توانیم مجموعه‌ای از وظایف را در یک تسک تعریف کنیم تا به راحتی و با زدن تنها یک دستور در ترمینال، کارها خودکار سازی شوند

```
// Run this task with : gulp
// OR gulp default
gulp.task('default', ['bower', 'icons', 'css', 'js']);
```



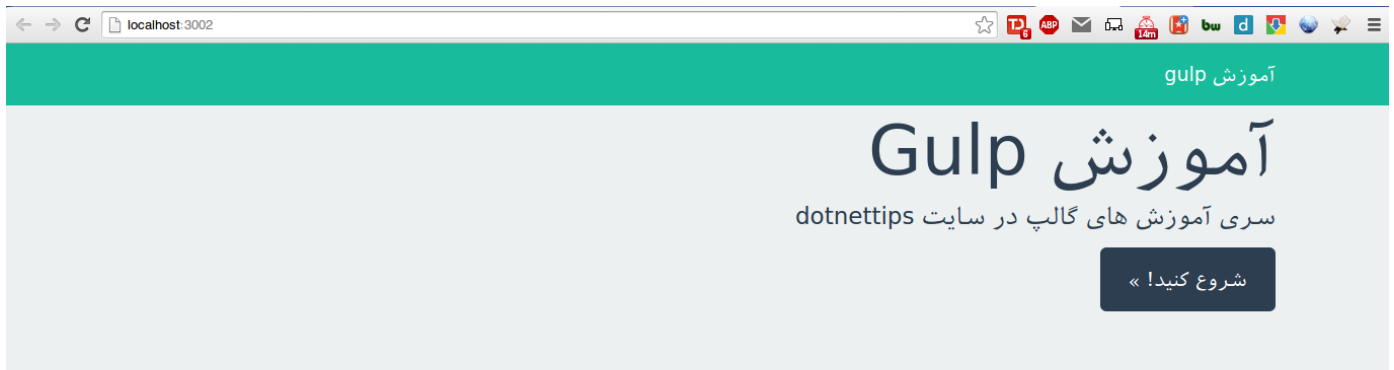
```
Terminal
mmdsharifi@mmdsharifi ~/Work/Gulp-RTLbootstrap-fontawesome $ gulp
[10:44:18] Using gulpfile ~/Work/Gulp-RTLbootstrap-fontawesome/gulpfile.js
[10:44:18] Starting 'bower'...
[10:44:18] Using cwd: /home/mmdsharifi/Work/Gulp-RTLbootstrap-fontawesome
[10:44:18] Using bower dir: ./bower_components
[10:44:18] Starting 'icons'...
[10:44:18] Starting 'css'...
[10:44:18] Starting 'js'...
[10:44:19] Finished 'icons' after 502 ms
[10:44:19] Finished 'js' after 520 ms
[10:44:19] Finished 'css' after 529 ms
[10:44:20] Finished 'bower' after 1.43 s
[10:44:20] Starting 'default'...
[10:44:20] Finished 'default' after 13 µs
mmdsharifi@mmdsharifi ~/Work/Gulp-RTLbootstrap-fontawesome $ gulp watch
[10:44:25] Using gulpfile ~/Work/Gulp-RTLbootstrap-fontawesome/gulpfile.js
[10:44:25] Starting 'watch'...
[10:44:25] Finished 'watch' after 6.71 ms
# wait for changes
```

بسیار خوب؛ ما توانستیم پایه‌ی ورک فلو یمان را بسازیم. در مقاله‌ی بعدی از پلاگین‌های دیگری برای بهینه‌سازی کارهایمان کمک خواهیم گرفت

[مخزن گیت هاب](#)

commit : [Update :gulp file and bower.json](#)

همانطور که در [مقاله ی قبلی](#) پایه ی ورک فلو ی خود را راه اندازی کردیم، در این مقاله می‌خواهیم با طراحی یک صفحه، با بوت استرپ شخصی سازی شده، در عمل با کارایی گالپ آشنا شویم.
دمو پایانی:



قسمت اول	قسمت دوم	قسمت سوم
Gulp ابزاری ست که شما را در انجام دادن کارهای مختلف توسعه ی وب، در سمت Front-end کمک می‌کند و اغلب برای کارهایی هم‌چون موارد ذیل بکار می‌رود:	در این مقاله و مقالات بعد می‌خواهیم کار خود را با راه اندازی یک workflow برای بوت استرپ، روند شخصی سازی آن را بسیار آسان و لذت بخش‌تر کنیم. ...	در قسمت اول گالپ را معرفی کردیم و در مقاله قبلی به نوشتن اولین تسک با گالپ پرداختیم. در این قسمت می‌خواهیم با نصب bower، پروژه ی workflow بوت استرپ راستچین شده را انجام دهیم.
شروع کنید! »	شروع کنید! »	شروع کنید! »

به هنگام سازی مرورگر و بارگذاری مجدد به صورت خودکار

یکی از موارد فوق العاده تکراری در هنگام توسعه ی وب، برای یک توسعه دهنده سمت کاربر (Front end Developer) ریلود کردن مرورگر است. همچنین تست وب سایت یا آپلود در موبایل و سایر دستگاه‌ها، متداول است. با پلاگین گالپ می‌توان این مشکل را به صورت بهینه‌ای حل کرد.

نصب

برای نصب دستور زیر را در مسیر پروژه، در ترمینال سیستم عامل خود وارد کنید.

```
npm install browser-sync gulp --save-dev
```

می‌دانیم برای استفاده از یک پلاگین باید توسط متد require آن را به یک متغیر انتساب دهیم؛ به صورت زیر:

```
var gulp = require('gulp'),
    sass = require('gulp-ruby-sass'),
    notify = require('gulp-notify'),
    browserSync = require('browser-sync'), // Add browser syncs plugin
    bower = require('gulp-bower');
```

توجه کنید هر پلاگینی که اضافه می‌کنید، باید تسک مربوط به آن را بنویسیم تا بتوانیم از آن استفاده کنیم. برای این پلاگین فقط مشخص کردن مسیر root سرور کافی است.

```
gulp.task('browserSync', function() {
  browserSync({
    server: {
      baseDir: './' //our server root
    }
  });
});
```

حال می‌خواهیم با زدن `gulp watch`، تمام کارهای ما به صورت خودکار انجام شوند. اما این دستور که در جلسه‌ی قبل آن‌را تعریف کردیم، فقط منتظر انجام یک تغییر است. تسک `watch` را به گونه‌ای تغییر می‌دهیم که ابتدا تسک‌های `brower sync`، `css`، `css` انجام شوند (به دلیل اینکه باید ابتدا، سرور راه اندازی شود) سپس گالپ منتظر تغییرات باشد و آنها را اعمال کند.

```
gulp.task('watch', [ 'css','browserSync'], function() {
})
```

تسک‌های `html,css,browserSync` قبل از تسک `watch` اجرا می‌شوند. طبق مستندات، این پلاگین یکی از توابع `API` [متد watch](#) است و کار آن همانند متد مشابهی در گالپ است. آن را برای ریلود خودکار مرورگر استفاده می‌کنیم.

```
// Rerun the task when a file changes
gulp.task('watch', ['html', 'css','browserSync'], function() {
  gulp.watch(config.sassPath + '/*.scss', ['css']);
  gulp.watch(config.htmlPath, ['html'] )
  browserSync.watch("./*.html").on("change", browserSync.reload); // browserSync watch task
});
```

می‌خواهیم بعد از کامپایل، فایل‌های `sass` هم مرورگر دوباره بارگذاری شوند. کد زیر را به انتهای تسک `css` اضافه می‌کنیم:

```
.pipe(browserSync.reload({
  stream: true
}));
```

بسیار خوب با انجام این کارها پلاگین باید به درستی کار کند.

شخصی سازی بوت استرپ

برای شخصی سازی بوت استرپ کافی است ابتدا فایل‌های `sass` بوت استرپ و `FontAwesome` را در `style.scss` ایمپورت کنیم؛ به این صورت:

```
@import "bootstrap";
@import "font-awesome";
```

حال دستور `gulp` را می‌زنیم. با اینکار فایل `style.scss` کامپایل می‌شود. می‌خواهیم یک فونت فارسی و یک قالب فلت را به پروژه‌امان اعمال کنیم. من فایل‌ها را اضافه کرده‌ام و شما با یک نگاه می‌توانید، چیزی را که گفتم درک کنید.

```
@import "fonts-fa";
@import "variable";
@import "bootstrap";
@import "font-awesome";
@import "rtl.scss";
@import "typography";
```

نکته : سعی کنید برای استایل هر قسمت، یک فایل مجزا درست کنید؛ مانند مثال بالا که در پروژه لحاظ شده. برای توسعه‌ی پروژه، ابتدا مخزن گیت هاب را فورک کرده و با زدن دستورات زیر کار خود را آغاز کنید:


```
sudo npm install
```

```
gulp
```

```
gulp watch
```

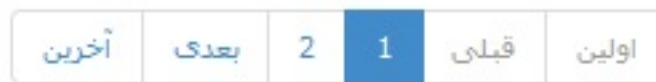
```
mmdsharifi@mmdsharifi ~/Work/Gulp-RTLbootstrap-fontawesome $ gulp watch
[12:58:37] Using gulpfile ~/Work/Gulp-RTLbootstrap-fontawesome/gulpfile.js
[12:58:37] Starting 'css'...
[12:58:37] Starting 'browserSync'...
[12:58:37] Finished 'browserSync' after 43 ms
[BS] Access URLs:
-----
    Local: http://localhost:3002
  External: http://192.168.1.7:3002
-----
    UI: http://localhost:3003
  UI External: http://192.168.1.7:3003
-----
[BS] Serving files from: ./
[12:58:39] Finished 'css' after 1.9 s
[12:58:39] Starting 'watch'...
[12:58:39] Finished 'watch' after 37 ms
[13:00:41] Starting 'css'...
[BS] 1 file changed (style.css)
[13:00:43] Finished 'css' after 1.62 s
[BS] File changed: index.html
[BS] File changed: index.html
[13:05:52] Starting 'css'...
[BS] 1 file changed (style.css)
[13:05:54] Finished 'css' after 1.83 s
[13:06:43] Starting 'css'...
[BS] 1 file changed (style.css)
[13:06:44] Finished 'css' after 1.69 s
[13:09:03] Starting 'css'...
```

[مخزن گیت هاب](#) : کامیت :

[Add : browserSync plugin and index.html](#)

بعد از استفاده از گریدهای JQGrid, Kendo و Grid.mvc مشکلاتی که با هر کدام از آنها داشتم، در نهایت به WebGrid که به صورت توکار وجود دارد، برای استفاده جهت نمایش اطلاعات رسیدم؛ از این جهت که به کتابخانه‌ی جانبی نیازی ندارد و از نظر سرعت و لود شدن بهینه می‌باشد، البته با اضافه کردن یکسری کدهای css. برای آشنایی بیشتر با این helper توصیه می‌کنم ابتدا این [مقاله](#) را مطالعه نمایید. به صورت پیش فرض WebGrid صفحه بندی را به صورت خیلی ساده فقط با نمایش اعداد و جهت نماهای جلو و عقب، نشان می‌دهد که برای پروژه‌های رسمی تا حدودی جالب نیست.

در این مطلب قصد داریم از کتابخانه‌ی bootstrap جهت صفحه بندی استفاده کنیم و در نهایت به صفحه بندی زیر برسیم:



برای اینکار، ابتدا قبل از هر چیزی به یک متد الحاقی برای انجام صفحه بندی سفارشی سازی شده، نیاز داریم که کدهای این متد به صورت زیر خواهد بود:

```
public static class WebGridExtensions
{
    public static HelperResult PagerList(
        this WebGrid webGrid,
        WebGridPagerModes mode = WebGridPagerModes.NextPrevious | WebGridPagerModes.Numeric,
        string firstText = null,
        string previousText = null,
        string nextText = null,
        string lastText = null,
        int numericLinksCount = 5)
    {
        return PagerList(webGrid, mode, firstText, previousText, nextText, lastText, numericLinksCount,
            explicitlyCalled: true);
    }

    private static HelperResult PagerList(
        WebGrid webGrid,
        WebGridPagerModes mode,
        string firstText,
        string previousText,
        string nextText,
        string lastText,
        int numericLinksCount,
        bool explicitlyCalled)
    {
        int currentPage = webGrid.PageIndex;
        int totalPages = webGrid.PageCount;
        int lastPage = totalPages - 1;

        var ul = new TagBuilder("ul");
        var li = new List<TagBuilder>();

        if (ModeEnabled(mode, WebGridPagerModes.FirstLast)) {
            if (String.IsNullOrEmpty(firstText)) {
                firstText = "اولین";
            }

            var part = new TagBuilder("li") {
                InnerHtml = GridLink(webGrid, webGrid.GetPageUrl(0), firstText)
            };
        }
    }
}
```

```

        if (currentPage == 0) {
            part.MergeAttribute("class", "disabled");
        }
        li.Add(part);
    }

    if (ModeEnabled(mode, WebGridPagerModes.NextPrevious)) {
        if (String.IsNullOrEmpty(previousText)) {
            previousText = "قبلی";
        }

        int page = currentPage == 0 ? 0 : currentPage - 1;

        var part = new TagBuilder("li") {
            InnerHtml = GridLink(webGrid, webGrid.GetPageUrl(page), previousText)
        };

        if (currentPage == 0) {
            part.MergeAttribute("class", "disabled");
        }

        li.Add(part);
    }

    if (ModeEnabled(mode, WebGridPagerModes.Numeric) && (totalPages > 1)) {
        int last = currentPage + (numericLinksCount / 2);
        int first = last - numericLinksCount + 1;
        if (last > lastPage) {
            first -= last - lastPage;
            last = lastPage;
        }
        if (first < 0) {
            last = Math.Min(last + (0 - first), lastPage);
            first = 0;
        }
        for (int i = first; i <= last; i++) {
            var pageText = (i + 1).ToString(CultureInfo.InvariantCulture);
            var part = new TagBuilder("li") {
                InnerHtml = GridLink(webGrid, webGrid.GetPageUrl(i), pageText)
            };

            if (i == currentPage) {
                part.MergeAttribute("class", "active");
            }

            li.Add(part);
        }
    }

    if (ModeEnabled(mode, WebGridPagerModes.NextPrevious)) {
        if (String.IsNullOrEmpty(nextText)) {
            nextText = "بعدی";
        }

        int page = currentPage == lastPage ? lastPage : currentPage + 1;

        var part = new TagBuilder("li") {
            InnerHtml = GridLink(webGrid, webGrid.GetPageUrl(page), nextText)
        };

        if (currentPage == lastPage) {
            part.MergeAttribute("class", "disabled");
        }

        li.Add(part);
    }

    if (ModeEnabled(mode, WebGridPagerModes.FirstLast)) {
        if (String.IsNullOrEmpty(lastText)) {
            lastText = "آخرین";
        }

        var part = new TagBuilder("li") {

```

```

        InnerHtml = GridLink(webGrid, webGrid.GetPageUrl(lastPage), lastText)
    };

    if (currentPage == lastPage) {
        part.MergeAttribute("class", "disabled");
    }

    li.Add(part);
}

ul.InnerHtml = string.Join("", li);

var html = "";
if (explicitlyCalled && webGrid.IsAjaxEnabled) {
    var span = new TagBuilder("span");
    span.MergeAttribute("data-swhgajax", "true");
    span.MergeAttribute("data-swhgcontainer", webGrid.AjaxUpdateContainerId);
    span.MergeAttribute("data-swhgcallback", webGrid.AjaxUpdateCallback);

    span.InnerHtml = ul.ToString();
    html = span.ToString();
} else {
    html = ul.ToString();
}

return new HelperResult(writer => {
    writer.Write(html);
});
}

private static String GridLink(WebGrid webGrid, string url, string text)
{
    TagBuilder builder = new TagBuilder("a");
    builder.SetInnerText(text);
    builder.MergeAttribute("href", url);
    if (webGrid.IsAjaxEnabled) {
        builder.MergeAttribute("data-swhglnk", "true");
    }
    return builder.ToString(TagRenderMode.Normal);
}

private static bool ModeEnabled(WebGridPagerModes mode, WebGridPagerModes modeCheck)
{
    return (mode & modeCheck) == modeCheck;
}
}

```

کلاس فوق باید در پوشه‌ی App_Code قرار گیرد.

پس از آن در View یی که اطلاعات را نمایش می‌دهید، فقط لازم است کد زیر را اضافه نمایید:

```

<div>
@grid.PagerList(mode: WebGridPagerModes.All)
</div>

```

تا اینجا متد مورد نظر برای انجام صفحه بندی گرید پیاده سازی شد. ادامه‌ی کار هم مشخص است؛ داشتن یک PartialView جهت نمایش لیست اطلاعات، پاس دادن دیتا به Partial و تمام.

در ادامه برای تکمیل بحث مثالی را از نحوه‌ی نمایش اطلاعات و صفحه بندی سفارشی نشان خواهیم داد:

PartialView لازم برای نمایش اطلاعات

تنظیمات لازم گرید :

```

@{
    WebGrid grid = new WebGrid(Model,
        rowsPerPage: 10,
        ajaxUpdateContainerId: "grid");
}

```

```
var rowIndex = ((grid.PageIndex + 1) * grid.RowsPerPage) - (grid.RowsPerPage - 1);
}
```

تعیین فیلدهای گرید :

```
@grid.Table(
    tableStyle: "table table-striped table-hover",
    headerStyle: "webgrid-header",
    alternatingRowStyle: "webgrid-alternating-row",
    selectedRowStyle: "webgrid-selected-row",
    rowStyle: "webgrid-row-style",
    columns: grid.Columns(
        grid.Column(columnName: "Name", header: "نام استان", style: "myfont"),
        grid.Column(columnName: "NameEn", header: "نام استان ( انگلیسی)", style: "myfont"),
        grid.Column(header: "", format: item => @Html.ActionLink("مدیریت شهرها", actionName:
MVC.Admin.City.ActionNames.Index, controllerName: MVC.Admin.City.Name, routeValues: new {Code=item.Code
}),htmlAttributes:null)),
        grid.Column(header: "",
            style: "text-align-center-col smallcell",
            format: item => @Html.ActionLink(linkText: "ویرایش", actionName: "Edit",
                controllerName: "Province", routeValues: new { area = "Admin", code =
item.Code },
                htmlAttributes: new { @class = "btn-sm btn-info vertical-center" })))
    ) )

<div>
    @grid.PagerList(mode: WebGridPagerModes.All)
</div>
```

خروجی حاصل به صورت زیر خواهد بود :

#	نام استان	نام استان (انگلیسی)		
۱	تهران	Tehan	مدیریت شهرها	ویرایش حذف
۲	کردستان	Kordistan	مدیریت شهرها	ویرایش حذف
۳	کرمانشاه	kermanshah	مدیریت شهرها	ویرایش حذف

اولین	قبلی	۱	۲	۳	۴	بعدی	آخرین
-------	------	---	---	---	---	------	-------

اگر طبق توضیحات بالا عمل کرده باشید، در نهایت صفحه بندی شما به صورت عمودی نمایش داده می شود؛ یعنی هر کدام از شماره صفحات در یک سطر. دلیل آن هم این است که تگ ul، کلاس pagination را ندارد. در کدهای بوت استراپ تعریف شده است که تمام li هایی که به صورت مستقیم داخل کلاس pagination هستند خصوصیات مورد نظر را بگیرند.

برای این کار دو راه حل وجود دارد :

راه حل اول: تغییر کدهای css

کدهای نوشته شده برای صفحه بندی در بوت استراپ را از حالت زیر:

```
.pagination > li
```

به حالت زیر تغییر دهید:

```
.pagination li
```

یادآوری: علامت > در CSS یعنی به صورت مستقیم و در شاخه‌ی اول.

راه حل دوم - افزودن کلاس .pagination به تگ ul:
ابتدا کلاس .pagination را از تگ div حذف نمایید:

```
<div >  
    @grid.PagedList(mode: WebGridPagerModes.All)  
</div>
```

و در کدهایی کلاس WebGridExtensions، در قسمتی که تگ ul اضافه می‌شود، کلاس مورد نظر را به آن اضافه می‌کنیم:

```
var ul = new TagBuilder("ul");  
ul.AddCssClass("pagination");
```

[دانلود کدهای این مثال](#)

نکته‌ای در مورد Webgrid

اگر نیاز داشتید به یکباره تمام اطلاعات را در گرید لود نکنید و به صورت n تا n رکوردها را نمایش دهید، در این حالت پس از پاس دادن لیستی از اطلاعات به View مورد نظر لازم است تعداد کل رکوردها را در یک متغیر به سمت View بفرستید. این کار به این دلیل می‌باشد که بتوان صفحه بندی را تولید کرد. برای این کار در بخش تنظیمات Webgrid مقدار source را برابر null قرار دهید و از قطعه کد زیر جهت بایند کردن گرید، بعد از کدهای تنظیمات WebGrid استفاده نمایید:

```
grid.Bind(Model, rowCount: (int)ViewBag.PageCount);
```

نظرات خوانندگان

نویسنده: امیر نوروزیان
تاریخ: ۱۴:۲۷ ۱۳۹۴/۰۸/۰۳

با سلام و تشکر از شما
هنگام استفاده از custom paging صفحه بندی مربوط به خود webgrid رو هم نشون میده که با غیر فعال کردن آن
canpage=false کلا صفحه بندی غیر فعال می‌شود.

<54321>

اولین	قبلی	2	3	4	بعدی	آخرین
-------	------	---	---	---	------	-------

نویسنده: عثمان رحیمی
تاریخ: ۱۵:۴۳ ۱۳۹۴/۰۸/۰۳

سلام .
لطفا از تابع Table به جای GetHtml استفاده کنید و خط زیر را هم از آن حذف کنید :

```
mode: WebGridPagerModes.All,
```

نمونه :

```
@{
    var grid = new WebGrid(
        source: null,
        canPage: true,
        rowsPerPage: 10,
        canSort: true,
        defaultSort: "Title"
    );
    grid.Bind(Model, rowCount: (int)ViewBag.PageCount, autoSortAndPage: false);
    var rowIndex = ((grid.PageIndex + 1) * grid.RowsPerPage) - (grid.RowsPerPage - 1);
}
@grid.Table(
    tableStyle: "table table-striped table-hover",
    headerStyle: "webgrid-header",
    footerStyle: "webgrid-footer",
    alternatingRowStyle: "webgrid-alternating-row",
    selectedRowStyle: "webgrid-selected-row",
    rowStyle: "webgrid-row-style",

    columns: grid.Columns(
        grid.Column(header: "#",
            style: "text-align-center-col",
            format: @<text>@(rowIndex++)</text>),
        grid.Column(columnName: "Title", header: "عنوان", style: "myfont"),
        grid.Column(columnName: "URL", header: "آدرس", style: "myfont"),

        grid.Column(header: "",
            style: "text-align-center-col smallcell",
            format: item => @Html.ActionLink(linkText: "ویرایش", actionName: "Edit",
                controllerName: "Link", routeValues: new { area = "Admin", id = item.Code
```

```

},
        htmlAttributes: new { @class = "btn-sm btn-info vertical-center" })),
    grid.Column(header: "",
        format: @<form action="Link/Delete/@item.Code" method="post">
            <input type="submit" class="btn-sm btn-danger submitlink"
                onclick="return confirm('آیا از حذف این آیتم مطمئن هستید ؟');"
                value="حذف" />
        </form>)))
<div class="text-center">
    @grid.PagedList(mode: WebGridPagerModes.All)
</div>

```

اگر شما تمام دیتا را به یکباره می‌خواهید نمایش دهید باید به صورت زیر تغییر دهید :

```

@{
    var grid = new WebGrid(
        source: Model,
        canPage: true,
        rowsPerPage: 10,
        canSort: true,
        defaultSort: "Title"
    );
    // grid.Bind(Model, rowCount: (int)ViewBag.PageCount, autoSortAndPage: false); // Delete this Line
    var rowIndex = ((grid.PageIndex + 1) * grid.RowsPerPage) - (grid.RowsPerPage - 1);
}

```