عنوان: **ایجاد پنجره های Bootstrap با HtmlHelper**
نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۳/۳۱ ۱:۲۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, jQuery, Twitter Bootstrap

چند وقت پیش لینکی را معرفی کردم که در آن به طراحی پنجره‌های بوت استرپ 3 با استفاده از جی کوئری پرداخته بود و از آنجا که من دوست دارم انعطاف بیشتری در استفاده از این مدل کتابخانه‌ها داشته باشم و مستندات آن را حفظ نکنم، آن‌ها را به HtmlHelper تبدیل می‌کنم.

ابتدا از این آدرس فایل‌های مورد نظر را دریافت کنید . دو عدد از آن‌ها 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 درفضای نام models جهت قرار دادن مدل‌ها 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 آیکن‌های بوت استراپ
Dialog.SpinIcon="";


//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 رویدا 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 بعدا 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 بعدا 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 می‌باشد. حدود 13 متد در آن پشتیبانی می‌شود که باعث می‌شود 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 می‌توانید در صورتیکه خواستید، در رویدادها و خارج از Html Helper از آن استفاده کنید.

نحوه‌ی استفاده از helper به شکل زیر است؛ فرض شده است که پنجره را تشکیل داده‌اید 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 آن‌ها 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، می‌توان از آن در بین کد رویدادها استفاده کرد. مثال زیر ایجاد یک رویداد، برای یکی از دکمه‌های پنجره است که با کلیک بر روی آن، پنجره بسته می‌شود:

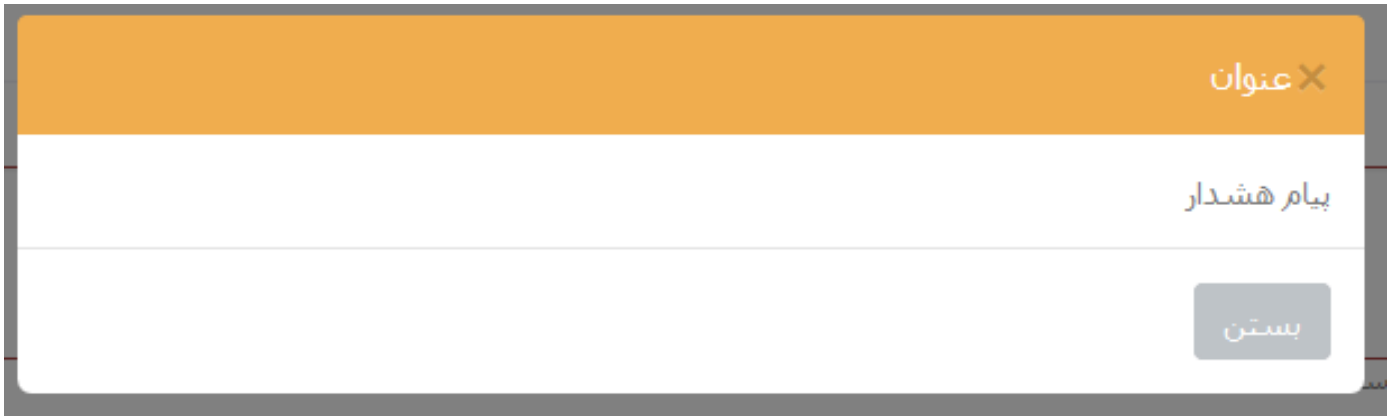
```
cancelButton.Action="function(){{{0}}}";
cancelButton.Action=string.format(cancelButton.Action,RunBootstrapDialogMethod("example1",BootstrapDial
ogMethods.Close));
```

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

```
@{
                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);
```



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.