

گروه بندی دکمه‌ها در Twitter bootstrap

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

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



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

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

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

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

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

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

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

مشخصات کاربر

نام

نام خانوادگی

نامشخص

خیر

بلی

فعال است؟

ارسال

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

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

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

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

در اینجا به همین نحو تعریف شده‌اند.

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

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

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

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

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

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

نظرات خوانندگان

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

با سلام، اگر ممکنه sample رو هم بذارین ممنون.

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

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

نویسنده: م.ح
تاریخ: ۱۱:۸ ۱۳۹۲/۰۴/۰۵

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

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

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

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