

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

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

نمایش اعلانات

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

نمایش اعلانات - کلاس alert-block

نمایش اعلانات - کلاس alert-danger

نمایش اعلانات - کلاس alert-error

نمایش اعلانات - کلاس alert-info

نمایش اعلانات - کلاس alert-success

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

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

```
<div>
نمایش اعلان
</div>
```

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

ابتدا کلاس‌های زیر را تعریف می‌کنیم:

کلاس Alert

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

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

کلاس AlertStyles

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

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

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

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

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

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

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

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

```

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

```

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

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

```

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

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

    return RedirectToAction("About");
}

```

: view

```

@{
    ViewBag.Title = "About";
}

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

```

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

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

```

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

```

```

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

```

نمایش پیام‌ها

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

```

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

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

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

```

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

```

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

```



خبر با عنوان خبر اول با موفقیت ذخیره گردید!



اطلاعات مورد نظر با موفقیت حذف گردید!

نظرات خوانندگان

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

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

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

به کار رفته است.

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

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

نویسنده: مهدی
تاریخ: ۱۱:۴۴ ۱۳۹۳/۰۴/۰۶

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

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

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

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

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

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

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

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

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

مرسی

با فقط خواندنی کردن سشن هم جواب نداد !

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

میمونه نوشتن پروایدر که باید براش وقت بذارم.