

کامپونت‌های jQuery زیادی وجود دارند که توسط آنها میتوان تصاویر را بصورت زمانبندی شده و به همراه افکت‌های زیبا در سایت خود نشان داد. مانند [اینجا](#).

در این قصد ایجاد helper برای کامپونت NivoSlider را داریم.



1- یک پروژه Asp.net Mvc 4.0 ایجاد میکنیم.

2- سپس فایل jquery.nivo.slider.pack.js ، فایل‌های css مربوط به این کامپونت و چهار تم موجود را از [سایت این کامپونت](#) و یا درون سورس مثال ارائه شده دریافت می‌کنیم.

3- به کلاسی BundleConfig رفته و کدهای زیر را اضافه میکنیم:

```
#region Nivo Slider
bundles.Add(new StyleBundle("~/Content/NivoSlider").Include("~/Content/nivoSlider/nivo-
slider.css"));
bundles.Add(new
```

```

StyleBundle("~/Content/NivoSliderDefaultTheme").Include("~/Content/nivoSlider/themes/default/default.css");
bundles.Add(new
StyleBundle("~/Content/NivoSliderDarkTheme").Include("~/Content/nivoSlider/themes/dark/dark.css"));
bundles.Add(new
StyleBundle("~/Content/NivoSliderLightTheme").Include("~/Content/nivoSlider/themes/light/light.css"));
bundles.Add(new
StyleBundle("~/Content/NivoSliderBarTheme").Include("~/Content/nivoSlider/themes/bar/bar.css"));
bundles.Add(new
ScriptBundle("~/bundles/NivoSlider").Include("~/Scripts/jquery.nivo.slider.pack.js"));

#endregion

```

سپس در فایل shared آنها را بصورت زیر اعمال میکنیم:

```

<!DOCTYPE html>
<html>
<head>
    <title>@ViewBag.Title</title>
    <script type="text/javascript" src="~/Scripts/jquery-1.9.1.min.js"></script>
    @Styles.Render("~/Content/NivoSlider",
    "~/Content/NivoSliderDefaultTheme", "~/Content/NivoSliderLightTheme")
    @Scripts.Render("~/bundles/NivoSlider")
</head>
<body>
    @RenderBody()
    @RenderSection("scripts", required: false)
</body>
</html>

```

4- یک پوشه با عنوان helper به پروژه اضافه میکنیم. سپس کلاسهای زیر را به آن اضافه میکنیم :

```

public class NivoSliderHelper
{
    #region Fields

    private string _id = "nivo1";
    private List<NivoSliderItem> _models = null;
    private string _width = "100%";
    private NivoSliderTheme _theme = NivoSliderTheme.Default;
    private string _effect = "random"; // Specify sets like= 'fold;fade;sliceDown'
    private int _slices = 15; // For slice animations
    private int _boxCols = 8; // For box animations
    private int _boxRows = 4; // For box animations
    private int _animSpeed = 500; // Slide transition speed
    private int _pauseTime = 3000; // How long each slide will show
    private int _startSlide = 0; // Set starting Slide (0 index)
    private bool _directionNav = true; // Next & Prev navigation
    private bool _controlNav = true; // 1;2;3... navigation
    private bool _controlNavThumbs = false; // Use thumbnails for Control Nav
    private bool _pauseOnHover = true; // Stop animation while hovering
    private bool _manualAdvance = false; // Force manual transitions
    private string _prevText = "Prev"; // Prev directionNav text
    private string _nextText = "Next"; // Next directionNav text
    private bool _randomStart = false; // Start on a random slide
    private string _beforeChange = ""; // Triggers before a slide transition
    private string _afterChange = ""; // Triggers after a slide transition
    private string _slideshowEnd = ""; // Triggers after all slides have been shown
    private string _lastSlide = ""; // Triggers when last slide is shown
    private string _afterLoad = "";

    #endregion

    string makeParameters()
    {
        var builder = new StringBuilder();
        builder.Append("{");
        builder.Append(string.Format("effect:{0}", _effect));
        builder.AppendLine(string.Format(",slices:{0}", _slices));
        builder.AppendLine(string.Format(",boxCols:{0}", _boxCols));
        builder.AppendLine(string.Format(",boxRows:{0}", _boxRows));
        builder.AppendLine(string.Format(",animSpeed:{0}", _animSpeed));
    }
}

```

```

        builder.AppendLine(string.Format(",pauseTime:{0}", _pauseTime));
        builder.AppendLine(string.Format(",startSlide:{0}", _startSlide));
        builder.AppendLine(string.Format(",directionNav:{0}", _directionNav.ToString().ToLower()));
        builder.AppendLine(string.Format(",controlNav:{0}", _controlNav.ToString().ToLower()));
        builder.AppendLine(string.Format(",controlNavThumbs:{0}",
_controlNavThumbs.ToString().ToLower()));
        builder.AppendLine(string.Format(",pauseOnHover:{0}", _pauseOnHover.ToString().ToLower()));
        builder.AppendLine(string.Format(",manualAdvance:{0}",
_manualAdvance.ToString().ToLower()));
        builder.AppendLine(string.Format(",prevText:'{0}'", _prevText));
        builder.AppendLine(string.Format(",nextText:'{0}'", _nextText));
        builder.AppendLine(string.Format(",randomStart:{0}", _randomStart.ToString().ToLower()));
        builder.AppendLine(string.Format(",beforeChange:{0}", _beforeChange));
        builder.AppendLine(string.Format(",afterChange:{0}", _afterChange));
        builder.AppendLine(string.Format(",slideshowEnd:{0}", _slideshowEnd));
        builder.AppendLine(string.Format(",lastSlide:{0}", _lastSlide));
        builder.AppendLine(string.Format(",afterLoad:{0}", _afterLoad));
        builder.Append(")");
        return builder.ToString();
    }
}

public NivoSliderHelper (
    string id,
    List<NivoSliderItem> models,
    string width = "100%",
    NivoSliderTheme theme = NivoSliderTheme.Default,
    string effect = "random",
    int slices = 15,
    int boxCols = 8,
    int boxRows = 4,
    int animSpeed = 500,
    int pauseTime = 3000,
    int startSlide = 0,
    bool directionNav = true,
    bool controlNav = true,
    bool controlNavThumbs = false,
    bool pauseOnHover = true,
    bool manualAdvance = false,
    string prevText = "Prev",
    string nextText = "Next",
    bool randomStart = false,
    string beforeChange = "function(){}",
    string afterChange = "function(){}",
    string slideshowEnd = "function(){}",
    string lastSlide = "function(){}",
    string afterLoad = "function(){}")
{
    _id = id;
    _models = models;
    _width = width;
    _theme = theme;
    _effect = effect;
    _slices = slices;
    _boxCols = boxCols;
    _boxRows = boxRows;
    _animSpeed = animSpeed;
    _pauseTime = pauseTime;
    _startSlide = startSlide;
    _directionNav = directionNav;
    _controlNav = controlNav;
    _controlNavThumbs = controlNavThumbs;
    _pauseOnHover = pauseOnHover;
    _manualAdvance = manualAdvance;
    _prevText = prevText;
    _nextText = nextText;
    _randomStart = randomStart;
    _beforeChange = beforeChange;
    _afterChange = afterChange;
    _slideshowEnd = slideshowEnd;
    _lastSlide = lastSlide;
    _afterLoad = afterLoad;
}

public IHtmlString GetHtml()
{
    var thm = "theme-" + _theme.ToString().ToLower();
    var sb = new StringBuilder();
    sb.AppendLine("<div style='width:" + _width + ";height:auto;margin:0px auto' class='" + thm
+ "'>");
    sb.AppendLine("<div id='" + _id + "' class='nivoSlider'>");
    foreach (var model in _models)
    {
        string img = string.Format("<img src='{0}' alt='{1}' title='{2}' />",

```

```

model.ImageFilePath, "", model.Caption);
        string item = "";
        if (model.LinkUrl.Trim().Length > 0 &&
            Uri.IsWellFormedUriString(model.LinkUrl, UriKind.RelativeOrAbsolute))
        {
            item = string.Format("<a href='{0}'>{1}</a>", model.LinkUrl, img);
        }
        else
        {
            item = img;
        }
        sb.AppendLine(item);
    }
    sb.AppendLine("</div>");
    sb.AppendLine("</div>");

    sb.AppendLine("<script type='text/javascript'>");
    //sb.AppendLine("$('#" + _id + "').parent().ready(function () {}");
    sb.Append("$(<document>).ready(function(){");
    sb.AppendLine("$('#" + _id + "').nivoSlider(" + makeParameters() + ");");
    //sb.AppendLine("$('nivo-controlNav a').empty();"); //semi hack for rtl layout
    //sb.AppendLine("$('#" + _id + "').nivoSlider();");
    sb.AppendLine("});");
    sb.AppendLine("</script>");

    return new HtmlString(sb.ToString());
}
}

```

```

public enum NivoSliderTheme
{
    Default, Light, Dark, Bar,
}

```

```

public class NivoSliderItem
{
    public string ImageFilePath { get; set; }
    public string Caption { get; set; }
    public string LinkUrl { get; set; }
}

```

6-سپس برای استفاده از این helper یک کنترلر به پروژه اضافه میکنیم مانند Home :

```

public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }
    public virtual ActionResult NivoSlider()
    {
        var models = new List<NivoSliderItem>
        {
            new NivoSliderItem()
            {
                ImageFilePath = Url.Content("~/Images/img1.jpg"),
                Caption = "عنوان اول",
                LinkUrl = "http://www.google.com",
            },
            new NivoSliderItem()
            {
                ImageFilePath = Url.Content("~/Images/img2.jpg"),
                Caption = "#htmlcaption",
                LinkUrl = "",
            },
            new NivoSliderItem()
            {
                ImageFilePath = Url.Content("~/Images/img3.jpg"),
                Caption = "عنوان سوم",
                LinkUrl = "",
            },
            new NivoSliderItem()
            {
                ImageFilePath = Url.Content("~/Images/img4.jpg"),
                Caption = "عنوان چهارم",
                LinkUrl = "",
            }
        }
    }
}

```

```

    },
    new NivoSliderItem()
    {
        ImageFilePath = Url.Content("~/Images/img5.jpg"),
        Caption = "عنوان پنجم",
        LinkUrl = "",
    }
};
return PartialView("_NivoSlider", models);
}
}

```

7- سپس ویوی Home را نیز ایجاد میکنیم:

```

@{
    ViewBag.Title = "Index";
}

<h2>Nivo Slider Index</h2>

@{ Html.RenderAction("NivoSlider", "Home");}

```

8- یک پارشال ویو نیز برای رندر کردن NivoSliderهای خود ایجاد میکنیم:

```

@using NivoSlider.Helper
@model List<NivoSliderItem>
@{
    var nivo1 = new NivoSliderHelper("nivo1", Model.Skip(0).Take(3).ToList(), theme:
    NivoSliderTheme.Default, width: "500px");
    var nivo2 = new NivoSliderHelper("nivo2", Model.Skip(3).Take(2).ToList(), theme:
    NivoSliderTheme.Light, width: "500px");
}
@nivo1.GetHtml()

<div id="htmlcaption">
    <strong>This</strong> is an example of a <em>HTML</em> caption with <a href="#">a link</a>.
</div>

<br />
<br />
@nivo2.GetHtml()

```

**نکته اول :** اگر بخواهیم بر روی تصویر موردنظر متنی نمایش دهیم کافیت خاصیت Caption را مقداردهی کنیم. برای نمایش توضیحاتی پیچیده‌تر (مانند نمایش یک div و لینک و غیره) بعنوان توضیحات یک تصویر خاص باید خاصیت Caption را بصورت "{نام عنصر}#" مقداردهی کنیم و ویژگی class مربوط به عنصر موردنظر را با nivo-html-caption مقداردهی کنیم. (همانند مثال بالا)

**نکته دوم :** این کامپوننت به همراه 4 تم (default- light- bar- dark) ارائه شده است. موقع استفاده از تم‌های دیگر باید رفرنس مربوط به آن تم را نیز اضافه کنید.

امیدوارم مفید واقع شود. [دریافت پروژه نمونه این مقاله](#)

## نظرات خوانندگان

نویسنده: bahman

تاریخ: ۱۱:۴۴ ۱۳۹۲/۰۲/۰۸

سلام تشکر میکنم. خیلی عالی بود.  
فقط یه مشکلی هست اونم نبودن WebGrease درون DLLهای شماست.  
چطوری میتونم پیدااش کنم؟

نویسنده: وحید نصیری

تاریخ: ۱۱:۴۷ ۱۳۹۲/۰۲/۰۸

این dll جزو فایل‌های استاندارد یک پروژه ASP.NET MVC 4 است. در اینجا پیوست نشده، چون نیازی نیست هربار تمام این فایل‌های تکراری را دریافت کنید.

فایل‌های آن در دو پوشه زیر بر روی سیستم شما موجود هستند: (اگر [ASP.NET MVC4](#) را نصب کرده باشید)

C:\Program Files\Microsoft ASP.NET\ASP.NET MVC 4\Packages

C:\Program Files\Microsoft ASP.NET\ASP.NET Web Pages\v 2.0\Packages

نویسنده: bahman

تاریخ: ۱۱:۵۴ ۱۳۹۲/۰۲/۰۸

مشکل با nuget حل شد.