

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:

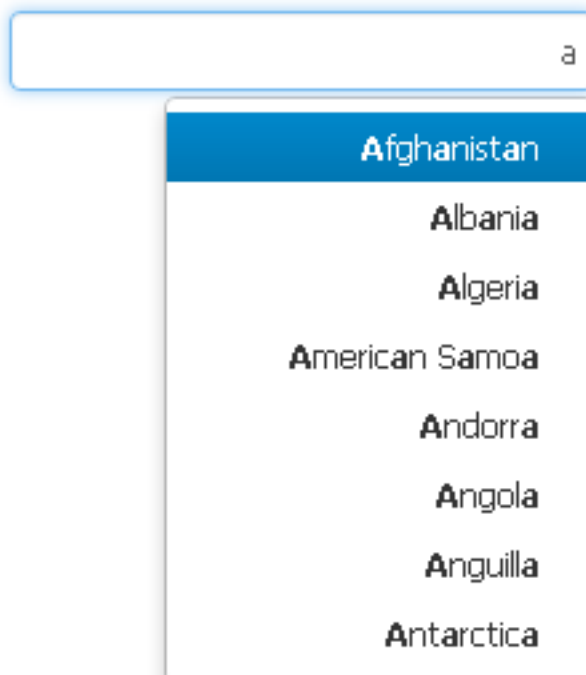
```
using System.Web.Mvc;
using System.Web.Script.Serialization;

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="["&quot;Afghanistan&quot;;&quot;Albania&quot;;&quot;Algeria&quot;;&quot;American
Samoa&quot;;&quot;Andorra&quot;;&quot;Angola&quot;;&quot;Anguilla&quot;;&quot;Antarctica&quot;;&quot;An
tigua and/or Barbuda&quot;]"
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، گروه‌بندی 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](#)