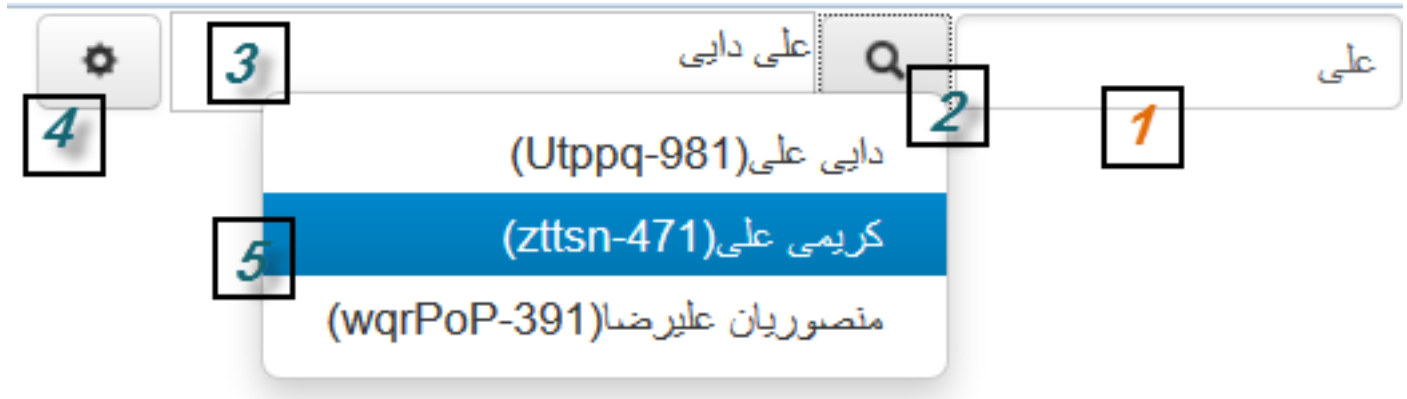


در این مطلب با نحوه استفاده از پلاگین جستجوی سفارشی searchboxmvc.js آشنا خواهید شد.



قبلاً در [اینجا](#) با نحوه ایجاد پلاگین jQuery آشنا شدید. روشی دیگری نیز برای ایجاد این نوع پلاگین‌ها وجود دارد و آن استفاده از [widget factory](#) موجود در پلاگین jQuery UI می‌باشد.

برای استفاده از این پلاگین که کدهای کامل آن در فایل پیوست موجود است، ابتدا باید فایل‌های لازم را به پروژه خود اضافه کنیم:

```
<link rel="stylesheet" href="@Url.Content("~/Content/bootstrap-rtl.css")" type="text/css" />
<script type="text/javascript" src="@Url.Content("~/scripts/jquery-2.0.2.min.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/jquery-ui-1.10.3.min.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/bootstrap-rtl.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/searchboxmvc.js")"></script>
```

سپس در کنترلر خود یک Action بصورت زیر ایجاد کنید:

```
[HttpPost]
public virtual ActionResult LoadData(string fieldName, string value, string stringFilterMode =
"startWith")
{
    Thread.Sleep(2000);
    var models = MakePersons();
    if (fieldName == "Id")
    {
        models = models.Where(p => p.Id == int.Parse(value)).Take(1).ToList();
    }
    else if (fieldName == "FirstName")
    {
        models = models.Where(p => p.FirstName.StartsWith(value)).ToList();
    }

    return Json(new { Status = "OK", Records = models });
}
private List<Person> MakePersons()
{
    var lst = new List<Person>();
    lst.Add(new Person() { Id = 1, Code = "Uytffs-098", FirstName = "احمد رضا", LastName =
"عابدزاده" });
    lst.Add(new Person() { Id = 2, Code = "fTuuuw-652", FirstName = "کریم", LastName = "باقری"
});
    lst.Add(new Person() { Id = 3, Code = "Lopapo-123", FirstName = "خداداد", LastName =
"عزیزی" });
    lst.Add(new Person() { Id = 4, Code = "Utppq-981", FirstName = "علی", LastName = "دایی" });
    lst.Add(new Person() { Id = 5, Code = "zttsn-471", FirstName = "علی", LastName = "کریمی"
});
}
```

```
lst.Add(new Person() { Id = 6, Code = "poiud-901", FirstName = "مهدی", LastName = "مهدوی" });
lst.Add(new Person() { Id = 7, Code = "wqrPoP-391", FirstName = "علیرضا", LastName = "منصوریان" });
return lst;
}
```

در ادامه در ویوی مورد نظر خود یک div ایجاد کنید. همین div خام با اعمال پلاگین بر روی آن ، بصورت یک پلاگین جستجو عمل خواهد کرد.

حال کدهای جاوا اسکریپت مورد نظر را برای اعمال پلاگین و تنظیمات مورد نیاز آن به div ایجاد شده می‌نویسیم:

```
<div id="div_SearchBoxContainer">
</div>
<script type="text/javascript">
$( "#div_SearchBoxContainer" ).searchboxmvc({
  loadUrl: '@Url.Action( "LoadData", controllerName: "Home" )',
  defaultStringFilterMode: "startWith",
  loadDataOnLeave: true,
  displayClass: "",
  displayNoResultClass: "",
  display: function ( element, record ) {
    $( element ).html( record.FirstName + " " + record.LastName );
  },
  listItemsDisplay: function ( element, record, index ) {
    return record.LastName + " " + record.FirstName + "( " + record.Code + " )";
  },
  fields: [
    {
      fieldName: "Id",
      fieldTitle: "شناسه",
      width: 100,
      defaultValueField: true
    },
    {
      fieldName: "FirstName",
      fieldTitle: "نام",
      width: 200,
      defaultDisplayField: true,
      filter: true,
      isStringType: true
    },
    {
      fieldName: "LastName",
      fieldTitle: "نام خانوادگی",
      filter: false,
      isStringType: true
    }
  ]
});
</script>
}
```

شرح پارامترهای افزونه searchboxmvc.js

loadUrl: آدرس اکشن متدی است که بصورت ajax ای فراخوانی شده و نتایج حاصل را بازگشت میدهد. نتایج حاصله باید با فرمت json بازگشت داده شوند. اگر نتایج موفقیت باشد باید بصورت `Json(new { Status = "OK", })` باشد. **Records = models** بازگشت داده شوند و اگر خطایی در این بین صورت گرفت مقدار Status نباید مقدار OK باشد. پارامترهای مورد نیاز این اکشن نیز باید به ترتیب با نام های `fieldName` و `value` باشند که `fieldName` نام فیلدی است که جستجو بر اساس آن صورت می‌گیرد و `value` همان مقدار وارد شده توسط کاربر است. **defaultStringFilterMode**: اگر فیلد مورد جستجو از نوع رشته ای باشد (یعنی `isStringType` آن برابر true باشد) آنگاه پارامتر سوم اکشن متد بطور خودکار مقداردهی خواهد شد. مقادیر این خاصیت میتواند `startWith` یا `contains` و یا `equal` باشد. **loadDataOnLeave**: اگر برابر false باشد، هر بار که متن input تغییر کرد بلافاصله یک تقاضا برای یافتن مقادیر به سرور فرستاده

میشود و نیازی نیست که فوکوس از کنترل خارج شود.

displayClass : نام کلاس css است که به 3 div اعمال خواهد شد.

displayNoResultClass : در صورتیکه جستجو نتیجه ای نداشته باشد این کلاس به 3 div اعمال خواهد شد.

display : یک فانکشن که برای ایجاد خروجی html برای نمایش در 3 div بکار می‌رود.

listItemsDisplay : یک فانکشن که برای ایجاد خروجی html برای آیتم‌ها بکار می‌رود.

fields : یک آرایه از فیلدهای موردنیاز پلاگین .

خاصیت‌های فیلد نیز بصورت زیر است:

fieldName : نام فیلد

fieldTitle : عنوان فیلد

defaultValueField : فیلد پیش فرض که جستجو بر اساس آن صورت می‌گیرد. اگر تعیین نشود فیلد اول آرایه به عنوان فیلد پیش فرض انتخاب خواهد شد.

defaultDisplayField : فیلد پیش فرض که برای نمایش متن 3 div بکار می‌رود(البته اگر پارامتر display تعیین نشود)

filter : اگر برابر true باشد این فیلد در لیست فیلدهای جستجو خواهد آمد و کاربر می‌تواند بر اساس آن جستجو انجام دهد.

isStringType : اگر برابر true باشد ، پارامتر سوم اکشن متد بطور خودکار مقداردهی خواهد شد.

لازم به ذکر است که این پلاگین کامل نیست و فقط برای ارائه مثال اینجا آورده شده است. هر یک از دوستان می‌توانند محتوای پلاگین را به سلیقه خود تغییر داده و پلاگین را کاملتر کنند.

[sample_mvc.zip](#)

نظرات خوانندگان

نویسنده: ح مراداف
تاریخ: ۱۶:۵۶ ۱۳۹۲/۱۱/۲۹

سلام ممنون از مطلب مفیدتون. چند تا سوال از خدمتتون داشتم: 1- آیا دیگه نیاز به قراردادن input نخواهیم داشت ؟ 2- فرض کنیم از این سیستم سرچ در بخشی از یک فرم استفاده کردیم و پس از کلیک روی دکمه submit نیاز داریم که به آی دی گزینه ای که توی بخش سرچ انتخاب شده ، دست پیدا کنیم ... چجوری باید به آی دی گزینه ای که توی کنترل سرچ انتخاب شده دست پیدا کنم ؟ با تشکر