

در این مطلب قصد داریم ترتیب عناصر نمایش داده شده توسط یک لیست را به کمک افزونه بسیار سبک وزن [jquery.sortable](#) تغییر داده و نتایج را در سمت سرور مدیریت کنیم. این افزونه بر اساس امکانات کشیدن و رها ساختن HTML5 کار می‌کند و با مرورگرهای IE8 به بعد سازگار است.

مدل‌های برنامه

```
using System.Collections.Generic;

namespace jQueryMvcSample05.Models
{
    public class Survey
    {
        public int Id { set; get; }
        public string Title { set; get; }

        public virtual ICollection<SurveyItem> SurveyItems { set; get; }
    }
}
```

```
namespace jQueryMvcSample05.Models
{
    public class SurveyItem
    {
        public int Id { set; get; }
        public string Title { set; get; }
        public int Order { set; get; }

        //[ForeignKey("SurveyId")]
        public virtual Survey Survey { set; get; }
        public int SurveyId { set; get; }
    }
}
```

به کمک این ساختار قصد داریم اطلاعات یک سیستم نظر سنجی را نمایش دهیم. تعدادی نظر سنجی به همراه گزینه‌های آن‌ها تعریف خواهند شد (یک رابطه one-to-many است). سپس توسط افزونه sortable می‌خواهیم ترتیب قرارگیری گزینه‌های آن‌را مشخص کنیم یا تغییر دهیم.

منبع داده فرضی برنامه

```
using System.Collections.Generic;
using jQueryMvcSample05.Models;

namespace jQueryMvcSample05.DataSource
{
    /// <summary>
    /// یک منبع داده فرضی جهت دموی ساده‌تر برنامه
    /// </summary>
    public static class SurveysDataSource
    {
        private static IList<Survey> _surveysCache;
        static SurveysDataSource()
        {
            _surveysCache = createSurveys();
        }

        public static IList<Survey> SystemSurveys
        {

```

```

        get { return _surveysCache; }
    }

    private static IList<Survey> createSurveys()
    {
        var results = new List<Survey>();
        for (int i = 1; i < 6; i++)
        {
            results.Add(new Survey
            {
                Id = i,
                Title = "نظر سنجی " + i,
                SurveyItems = new List<SurveyItem>
                {
                    new SurveyItem{ Id = 1, SurveyId = i, Title = "گزینه 1", Order = 1 },
                    new SurveyItem{ Id = 2, SurveyId = i, Title = "گزینه 2", Order = 2 },
                    new SurveyItem{ Id = 3, SurveyId = i, Title = "گزینه 3", Order = 3 },
                    new SurveyItem{ Id = 4, SurveyId = i, Title = "گزینه 4", Order = 4 }
                }
            });
        }
        return results;
    }
}

```

در اینجا نیز از یک منبع داده فرضی تشکیل شده در حافظه جهت سهولت دموی برنامه استفاده خواهد شد.

کدهای کنترلر برنامه

```

using System.Collections.Generic;
using System.Linq;
using System.Web.Mvc;
using System.Web.UI;
using jQueryMvcSample05.DataSource;
using jQueryMvcSample05.Security;

namespace jQueryMvcSample05.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            var surveysList = SurveysDataSource.SystemSurveys;
            return View(surveysList);
        }

        [HttpPost]
        [AjaxOnly]
        [OutputCache(Location = OutputCacheLocation.None, NoStore = true)]
        public ActionResult SortItems(int? surveyId, string[] items)
        {
            if (items == null || items.Length == 0 || surveyId == null)
                return Content("nok");

            updateSurvey(surveyId, items);

            return Content("ok");
        }

        /// <summary>
        /// این متد جهت آشنایی با پروسه به روز رسانی ترتیب گزینه‌ها در اینجا قرار گرفته است
        /// بدیهی است محل قرارگیری آن باید در لایه سرویس برنامه اصلی باشد
        /// </summary>
        private static void updateSurvey(int? surveyId, string[] items)
        {
            var itemIds = new List<int>();
            foreach (var item in items)
            {
                itemIds.Add(int.Parse(item.Replace("item-row-", string.Empty)));
            }

            var survey = SurveysDataSource.SystemSurveys.FirstOrDefault(x => x.Id == surveyId.Value);
            if (survey == null)

```

```

        return;

        int order = 0;
        foreach (var itemId in itemIds)
        {
            order++;
            var surveyItem = survey.SurveyItems.FirstOrDefault(x => x.Id == itemId);
            if (surveyItem == null) continue;
            surveyItem.Order = order;
        }

        //todo: call save changes ....
    }
}

```

و کدهای View متناظر

```

@model IList<jQueryMvcSample05.Models.Survey>
@{
    ViewBag.Title = "Index";
    var sortUrl = Url.Action(actionName: "SortItems", controllerName: "Home");
}
<h2>
    نظر سنجی‌ها
</h2>
@foreach (var survey in Model)
{
    <fieldset>
        <legend>@survey.Title</legend>
        <div id="sortable-@survey.Id">
            @foreach (var surveyItem in survey.SurveyItems.OrderBy(x => x.Order))
            {
                <div id="item-row-@surveyItem.Id">
                    <span class="handles">:::</span>
                    @surveyItem.Title
                </div>
            }
        </div>
    </fieldset>
}
<div>
    لطفا برای تغییر ترتیب آیتم‌های تعریف شده، از امکان کشیدن و رها کردن تعریف شده بر
    روی آیکن‌های :: در کنار هر آیتم استفاده نمائید.
</div>
@section JavaScript
{
    <script type="text/javascript">
        $(document).ready(function () {
            $('div[id^="sortable"]').sortable({ handle: 'span' }).bind('sortupdate', function (e, ui) {
                var sortableItemId = $(ui.item).parent().attr('id');
                var surveyId = sortableItemId.replace('sortable-', '');
                var items = [];
                $('#'+ sortableItemId + ' div').each(function () {
                    items.push($(this).attr('id'));
                });
                //alert(items.join('&'));
                $.ajax({
                    type: "POST",
                    url: "@sortUrl",
                    data: JSON.stringify({ items: items, surveyId: surveyId }),
                    contentType: "application/json; charset=utf-8",
                    dataType: "json",
                    complete: function (xhr, status) {
                        var data = xhr.responseText;
                        if (xhr.status == 403) {
                            window.location = "/login";
                        } else if (status === 'error' || !data || data == "nok") {
                            alert('خطایی رخ داده است');
                        }
                        else {
                            alert('انجام شد');
                        }
                    }
                });
            });
        });
    </script>
}

```

```
});
</script>
}
```

توضیحات

در اینجا نیاز بود تا ابتدا کدهای کنترلر و View ارائه شوند، تا بتوان در مورد ارتباطات بین آن‌ها بهتر بحث کرد. در ابتدای نمایش صفحه Home، رکوردهای نظرسنجی‌ها از منبع داده دریافت شده و به View ارسال می‌شوند. در View برنامه یک حلقه تشکیل گردیده و این موارد رندر خواهند شد.

هر نظر سنجی با یک div بیرونی که با id مساوی sortable شروع می‌شود، آغاز گردیده و گزینه‌های آن نظر سنجی نیز توسط divهایی با id مساوی item-row شروع خواهند گردید. هر کدام از این idها حاوی id رکوردهای متناظر هستند. از این idها در کدهای برنامه جهت یافتن یک نظر سنجی یا یک ردیف مشخص برای به روز رسانی ترتیب آن‌ها استفاده خواهیم کرد. ادامه کار، به تنظیمات و اعمال افزونه sortable مرتبط می‌شود. توسط تنظیم ذیل به jQuery اعلام خواهیم کرد، هر جایی یک div با id شروع شده با sortable یافتی، افزونه sortable را به آن متصل کن:

```
$('#div[id^="sortable"]').sortable
```

در ادامه در ناحیه و div ایی که عمل کشیدن و رها شدن رخ داده، id این div را بدست آورده و سپس کلیه row-itemهای آن را در آرایه‌ای به نام items قرار می‌دهیم:

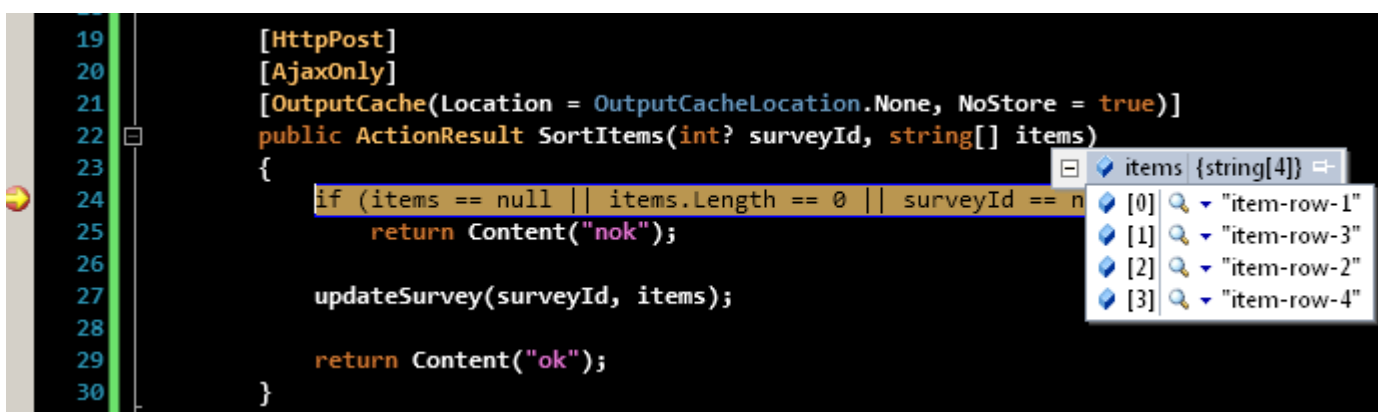
```
var sortableItemId = $(ui.item).parent().attr('id');
var surveyId = sortableItemId.replace('sortable-', '');
var items = [];
$('#' + sortableItemId + ' div').each(function () {
    items.push($(this).attr('id'));
});
```

اکنون که به id یک نظر سنجی و همچنین idهای ردیف‌های مرتب شده حاصل دسترسی داریم، آن‌ها را توسط jQuery Ajax کنترلر برنامه ارسال می‌کنیم:

```
data: JSON.stringify({ items: items, surveyId: surveyId })
```

امضای اکشن متد SortItems نیز دقیقاً بر همین مبنا تنظیم شده است:

```
public ActionResult SortItems(int? surveyId, string[] items)
```



اطلاعاتی که در اینجا دریافت می‌شوند در متد updateSurvey مورد استفاده قرار خواهند گرفت. بر اساس surveyId دریافتی،

نظرسنجی مرتبط را یافته و سپس به گزینه‌های آن دست خواهیم یافت. اکنون نوبت به پردازش آرایه items دریافت شده است. این آرایه بر اساس انتخاب کاربر مرتب شده است.

دریافت کدها و پروژه کامل این قسمت

[jQueryMvcSample05.zip](#)

نظرات خوانندگان

نویسنده: علی حق جو
تاریخ: ۱۳۹۲/۰۲/۳۰ ۱۰:۴۵

با سلام.
امکان مرتب سازی یک table با این کامپوننت وجود دارد؟
سعی من بی نتیجه بود. با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۳۰ ۱۱:۱۱

بهتره به issue tracker مخصوص آن مراجعه کنید. [یک نمونه از کار با جدول](#)