عنوان: **فرمت کردن اطلاعات نمایش داده شده به کمک jqGrid در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۲ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

پیشنیاز این بحث مطالعه‌ی مطلب « صفحه بندی و مرتب سازی خودکار اطلاعات به کمک jqGrid در ASP.NET MVC » است و در اینجا جهت کوتاه شدن بحث، صرفا به تغییرات مورد نیاز جهت اعمال بر روی مثال اول اکتفاء خواهد شد.

صورت مساله

می‌خواهیم اطلاعات نمایش داده شده در گرید را به نحوی فرمت کنیم که
الف) اگر id ردیف مساوی 5 بود، رنگ و پس زمینه‌ی آن تغییر کند.
ب) نام محصول، به جزئیات آن لینک شود و این اطلاعات توسط یک jQuery UI Dialog نمایش داده شود.
ج) عدد قیمت با سه رقم جدا کننده همراه باشد.

تکمیل مدل برنامه

مدل قسمت اول صرفا یک محصول بود. مدل قسمت جاری، اطلاعات تولید/تامین کننده آن‌را توسط کلاس Supplier نیز به همراه دارد:

```
namespace jqGrid02.Models
{
    public class Product
    {
        public int Id { set; get; }
        public string Name { set; get; }
```

```
        public decimal Price { set; get; }
        public Supplier Supplier { set; get; }
    }

    public class Supplier
    {
public int Id { set; get; }
        public string CompanyName { set; get; }
        public string Address { set; get; }
        public string PostalCode { set; get; }
        public string City { set; get; }
        public string Country { set; get; }
        public string Phone { set; get; }
        public string HomePage { set; get; }
    }
}
```

کدهای سمت سرور

کدهای سمت سرور مانند متد GetProducts به همراه صفحه بندی و مرتب سازی پویای آن دقیقا مانند قسمت قبل است.
در اینجا فقط یک اکشن متد جدید جهت بازگشت اطلاعات تولید کننده‌ای مشخص با فرمت JSON، اضافه شده‌است:

```
public ActionResult GetGetSupplierData(int id)
        {
            var list = ProductDataSource.LatestProducts;
            var product = list.FirstOrDefault(x => x.Id == id);
            if (product == null)
                return Json(null, JsonRequestBehavior.AllowGet);

            return Json(new
                        {
                            product.Supplier.CompanyName,
                            product.Supplier.Address,
                            product.Supplier.PostalCode,
                            product.Supplier.City,
                            product.Supplier.Country,
                            product.Supplier.Phone,
                            product.Supplier.HomePage
                        }, JsonRequestBehavior.AllowGet);
        }
```

کدهای سمت کلاینت

صفحه دیالوگی که قرار است اطلاعات تولید کننده را نمایش دهد، یک چنین ساختاری دارد:

```
<div dir="rtl" id="supplierDialog">
    <span id="CompanyName"></span><br /><br />
    <span id="Address"></span><br />
    <span id="PostalCode"></span>, <span id="City"></span><br />
    <span id="Country"></span><br /><br />
    <span id="Phone"></span><br />
    <span id="HomePage"></span>
</div>
```

و تغییرات گرید برنامه به شرح زیر است:

```
<script type="text/javascript">
        function showSupplierDialog(linkElement, supplierId) {
            //request json data
            $.getJSON('@Url.Action("GetGetSupplierData","Home")', { id: supplierId }, function (data) {
                //set values in dialog
                for (var property in data) {
                    if (data.hasOwnProperty(property)) {
                        $('#' + property).text(data[property]);
                    }
                }

                //get link position
                var linkPosition = $(linkElement).offset();
```

```
                $('#supplierDialog').dialog('option', 'position', [linkPosition.left,
linkPosition.top]);
                //open dialog
                $('#supplierDialog').dialog('open');
            });
        }

        $(document).ready(function () {

            $('#supplierDialog').dialog({
                 autoOpen: false, bgiframe: true, resizable: false, title: 'تولید کننده'
            });

            $('#list').jqGrid({
                // .... مانند قبل
                colNames: ['قیمت' ,'نام محصول' ,'شماره'],
                //columns model
                colModel: [
                    {
                        name: 'Id', index: 'Id', align: 'right', width: 20,
                        formatter: function (cellvalue, options, rowObject) {
                            var cellValueInt = parseInt(cellvalue);
                            if (cellValueInt == 5) {
                                return "<span style='background: brown; color: yellow'>" + cellvalue +
"</span>";
                            }
                            return cellvalue;
                        }
                    },
                    {
                        name: 'Name', index: 'Name', align: 'right', width: 300,
                        formatter: function (cellvalue, options, rowObject) {
                            return "<a href='#' onclick='showSupplierDialog(this, " + rowObject[0] +
");'>" + cellvalue + "</a>";
                        }
                    },
                    {
                        name: 'Price', index: 'Price', align: 'center', width: 50,
                        formatter: 'currency',
                        formatoptions:
                        {
                            decimalSeparator: '.', thousandsSeparator: ',', decimalPlaces: 2, prefix:
'$'
                        }
                    }
                ],
                // .... مانند قبل
            });
        });
    </script>
```

- همانطور که ملاحظه می‌کنید، توسط خاصیت formatter می‌توان عناصر در حال نمایش را فرمت کرد و بر روی نحوه‌ی نمایش نهایی آن‌ها تاثیرگذار بود.

در حالت ستون Id، از یک formatter سفارشی استفاده شده‌است. در اینجا این فرمت کننده به صورت یک callback عمل کرده و پیش از رندر نهایی اطلاعات، مقدار سلول جاری را توسط cellvalue در اختیار ما قرار می‌دهد. در این بین هر نوع فرمتی را که نیاز است می‌توان اعمال کرد و سپس یک رشته را بازگشت می‌دهیم. این رشته در سلول جاری درج خواهد شد.

- اگر مانند ستون Name، نیاز به مقادیر سایر سلول‌ها نیز وجود داشت، می‌توان از آرایه‌ی rowObject استفاده کرد. برای مثال در این حالت، یک لینک که کلیک بر روی آن سبب فراخوانی تابع showSupplierDialog می‌شود، در سلول‌های ستون Name درج خواهند شد. اولین rowObject که در اینجا مورد استفاده است، به ستون اول یا همان Id محصول اشاره می‌کند.

- در ستون Price از یک سری formatter از پیش تعریف شده استفاده شده‌است. نمونه‌ای از آن را در قسمت اول در ستون نمایش وضعیت موجود بودن محصول با تنظیم formatter: checkbox مشاهده کرده‌اید. در اینجا از یک formatter توکار دیگر به نام currency برای کار با مقادیر پولی استفاده شده‌است به همراه تنظیمات خاص آن.

- متد showSupplierDialog طوری تنظیم شده‌است که پس از دریافت Id یک محصول، آن‌را به سرور ارسال کرده و مشخصات تولید کننده‌ی آن‌را با فرمت JSON دریافت می‌کند. سپس در حلقه‌ای که مشاهده می‌کنید، خواص شیء جاوا اسکریپتی دریافتی استخراج و به spanهای supplierDialog انتساب داده می‌شوند. جهت سهولت کار، Id این spanها دقیقا مساوی Id خواص شیء دریافتی از سرور، درنظر گرفته شده‌اند.

- در مورد راست به چپ نمایش داده شدن عنوان دیالوگ، تغییرات CSS ایی لازم است که در قسمت اول بیان شدند.

برای مطالعه بیشتر

لیست کامل فرمت کننده‌های توکار

فرمت کننده‌های سفارشی

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

jqGrid02.zip