عنوان: **فعال سازی و پردازش جستجوی پویای jqGrid در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۳ 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.

تغییرات مورد نیاز سمت کلاینت جهت فعال سازی جستجو در jqGrid

در سمت کلاینت، در حین تعریف ستون‌ها، ابتدا باید توسط مقدار دهی خاصیت search، ستون‌های مشارکت کننده‌ی در حین جستجو را مشخص کرد:

```
colModel: [
                    {
                        name: 'Name', index: 'Name', align: 'right', width: 200,
                        search: true, stype: 'text', searchoptions: { sopt: searchOptions }
                    },
                    {
                        name: 'Supplier.Id', index: 'Supplier.Id', align: 'right', width: 200,
                        search: true, stype: 'select', edittype: 'select', searchOperators: true,
                        searchoptions:
                        {
                            sopt: searchOptions, dataUrl: '@Url.Action("SuppliersSelect","Home")'
                        }
                    }
                ],
```

- برای نمونه در اینجا search: true، جهت دو ستون نام محصول و نام تولید کننده، تنظیم شده‌اند.
- stype، روش مقایسه‌ی مقادیر را مشخص می‌کند. مقدار پیش فرض آن text است و مقادیری مانند int، integer، float، number، numeric، date و datetime را می‌پذیرد.
- searchoptions برای تنظیم جزئیات نحوه‌ی جستجوی بر روی فیلدها بکار می‌رود. توسط sopt می‌توان آرایه‌ای با مقادیر ذیل را مقدار دهی کرد:

```
var searchOptions = ['eq', 'ne', 'lt', 'le', 'gt', 'ge', 'bw', 'bn', 'in', 'ni', 'ew', 'en', 'cn',
'nc'];
```

eq به معنای مساوی است، ne، مساوی نیست، lt کمتر و به همین ترتیب.
البته باید دقت داشت که آرایه فوق کاملترین حالت ممکن است و ضرورتی ندارد تمام حالات را برای یک فیلد تعریف کرد. چون برای مثال جستجو cn یا contains برای مقادیر رشته‌ای معنا دارد و نه سایر حالات.
- searchoptions گزینه‌های دیگری را نیز می‌تواند شامل شود. برای مثال در حین نمایش جستجوی داخل ردیفی یا صفحه‌ی دیالوگ مخصوص آن، قصد داریم فیلد نام تولید کننده را توسط یک drop down نمایش دهیم و نه یک text box پیش فرض. برای این منظور dataUrl مقدار دهی شده‌است.
SuppliersSelect آن به اکشن متد ذیل اشاره می‌کند که لیست تولید کنندگان را با فرمت لیستی از SelectListItemها به یک partial view تولید کننده‌ی دراپ داون ارسال می‌کند:

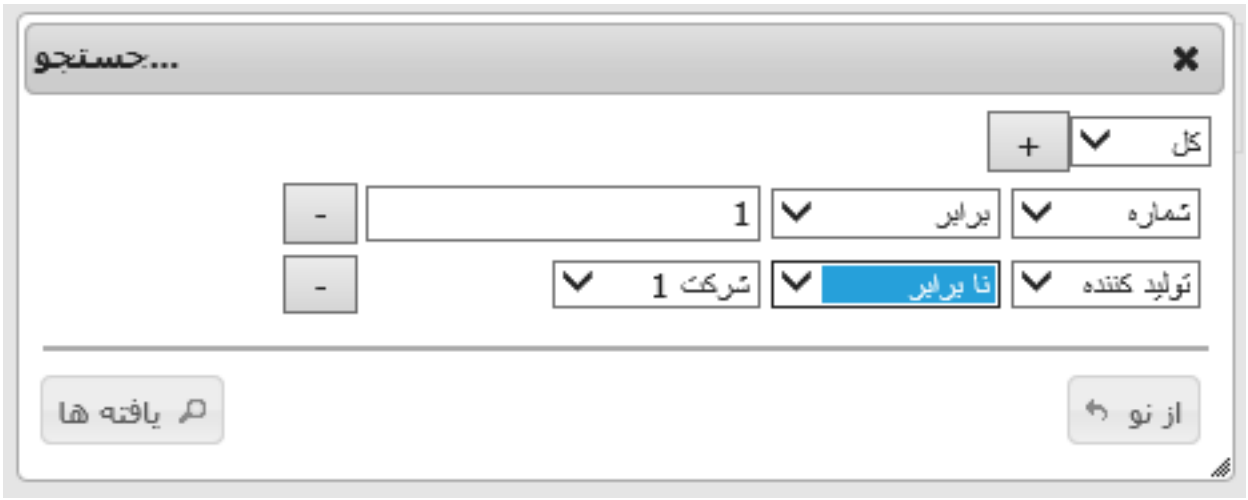
```
public ActionResult SuppliersSelect()
        {
            var list = ProductDataSource.LatestProducts;
            var suppliers = list.Select(x => new SelectListItem
            {
                Text = x.Supplier.CompanyName,
                Value = x.Supplier.Id.ToString(CultureInfo.InvariantCulture)
            }).ToList();
            return PartialView("_SelectPartial", suppliers);
        }
```

و محتوای فایل _SelectPartial نیز به صورت ذیل است:

```
@model IList<SelectListItem>
@Html.DropDownList("srch", Model)
```

در این حالت با نمایش صفحه‌ی جستجو و انتخاب فیلد نام تولید کننده، به صورت خودکار یک dorp down پویا نمایش داده خواهد شد.

- اگر دقت کرده باشید، نام فیلد تولید کننده با Supplier.Id مقدار دهی شده‌است. علت اینجا است که در زمان استفاده از drop down، مقدار Id آیتم انتخابی، به سرور ارسال می‌شود. به همین جهت کار کردن پویا با Supplier.Id ساده‌تر خواهد بود.



- برای نمایش دیالوگ و یا نوار ابزار توکار جستجو، می‌توان دکمه‌هایی را به فوتر گرید اضافه کرد:


```
$(document).ready(function () {
            $('#list').jqGrid({
              // ... مانند قبل و توضیحات فوق
            })
            .jqGrid('navGrid', '#pager',
                { add: false, edit: false, del: false },
                {},  // default settings for edit
                {},  // default settings for add
                {},  // delete instead that del:false we need this
                {
                    // search options
                    multipleSearch: true,
                    closeOnEscape: true,
                    closeAfterSearch: true,
                    ignoreCase: true
                })
            .jqGrid('navButtonAdd', "#pager", {
                caption: "نوار ابزار جستجو", title: "Search Toolbar", buttonicon: 'ui-icon-search',
                onClickButton: function () {
                    toolbarSearching();
                }
            });
        });


        function toolbarSearching() {
            $('#list').filterToolbar({
                groupOp: 'OR',
                defaultSearch: "cn",
                autosearch: true,
                searchOnEnter: true,
                searchOperators: true, // فعال سازی منوی اپراتورها
                stringResult : true // وجود این سطر سبب می‌شود تا اپراتورها به سرور ارسال شوند
            });
        };

        function singleSearching() {
            $('#list').searchGrid({
                closeAfterSearch: true
            });
        };

        function advancedSearching() {
            $('#list').searchGrid({
                multipleSearch: true,
                closeAfterSearch: true
            });
        };
```

در اینجا نکات ذیل قابل توجه هستند:

- با استفاده از متد jqGrid و پارامتر navGrid، در ناحیه‌ی pager گرید، تنظیمات جستجو را فعال کرده‌ایم.

multipleSearch به معنای امکان جستجوی همزمان بر روی بیش از یک فیلد است. closeOnEscape سبب بسته شدن صفحه‌ی دیالوگ جستجو با فشردن دکمه‌ی ESC می‌شود. اگر closeAfterSearch به true تنظیم نشود، صفحه‌ی دیالوگ جستجو پس از جستجو، در صفحه باقی مانده و بسته نخواهد شد.

- این دکمه‌ی جستجو، جزو موارد توکار jqGrid است. اگر قصد داشته باشیم یک دکمه‌ی سفارشی دیگر را نیز اضافه کنیم، مجددا از متد jqGrid با پارامتر navButtonAdd در ناحیه‌ی pager استفاده خواهیم کرد. کلیک بر روی آن سبب اجرای متد

toolbarSearching می‌شود.

در اینجا حداقل سه نوع جستجو را می‌توان فعال کرد:
- filterToolbar که سبب نمایش نوار ابزار جستجو، دقیقا بالای ستون‌های جدول می‌شود.
- searchGrid که صفحه‌ی دیالوگ مستقلی را جهت جستجو به صورت پویا تولید می‌کند. اگر خاصیت multipleSearch آن به true تنظیم نشود، این جستجو هربار تنها بر روی یک فیلد قابل انجام خواهد بود و برعکس.
- در حالت جستجوی نوار ابزاری، اگر خواص searchOperators و stringResult به true تنظیم شوند، مانند تصویر ذیل، به ازای هر ستون می‌توان از عملگرهای مختلفی استفاده کرد. در غیراینصورت جستجوی انجام شده بر اساس groupOp و defaultSearch پیش فرض انجام می‌شود. یعنی And یا Or تمام موارد تنها در حالت مثلا contains یا تساوی و امثال آن و نه حالت پیشرفته‌ی انتخاب عملگرها توسط کاربر.

یک نکته

اگر می‌خواهید صفحه‌ی جستجو در وسط صفحه ظاهر شود، می‌توانید از تنظیمات CSS ذیل استفاده کنید:

```
/* align center search popup in jqgrid */
.ui-jqdialog {
    display: none;
    width: 300px;
    position: absolute;
    padding: .2em;
    font-size: 11px;
    overflow: visible;
    left: 30% !important;
    top: 40% !important;
}
```

پردازش سمت سرور جستجوی پویای jqGrid

کدهای سمت سرور، با کدهای استفاده از dynamic LINQ مایکروسافت یکی است. با این تفاوت که اینبار قسمت where این کوئری نیز پویا می‌باشد. پیشتر قسمت order by را پویا پردازش کرده بودیم. برای ساخت where پویا که در dynamic LINQ به خوبی

پشتیبانی می‌شود، باید ابتدا ساختار اطلاعات ارسالی به سرور را آنالیز کنیم:

```
//single field search
//_search=true&nd=1403935889318&rows=10&page=1&sidx=Id&sord=asc&searchField=Id&searchString=4444&search
Oper=eq&filters=

//multi-field search
//_search=true&nd=1403935941367&rows=10&page=1&sidx=Id&sord=asc&filters=%7B%22groupOp%22%3A%22AND%22%2C
%22rules%22%3A%5B%7B%22field%22%3A%22Id%22%2C%22op%22%3A%22eq%22%2C%22data%22%3A%2244%22%7D%2C%7B%22fie
ld%22%3A%22SupplierID%22%2C%22op%22%3A%22eq%22%2C%22data%22%3A%221%22%7D%5D%7D&searchField=&searchStrin
g=&searchOper=
// filters ->
{"groupOp":"AND","rules":[{"field":"All","op":"cn","data":"fffff"},{"field":"Price","op":"bn","data":"f
fff"}]}

//toolbar search
//_search=true&nd=1403935593036&rows=10&page=1&sidx=Id&sord=asc&Id=2&Name=333&SupplierID=1&CategoryID=1
&Price=44
```

در اینجا ساختار ارسالی به سرور را در سه حالت مختلف جستجوی پویای jqGrid، ملاحظه می‌کنید:

- در تمام این حالات پارامتر _search مساوی true است (تفاوت آن با درخواست اطلاعات معمولی).
- در حالت جستجوی نوار ابزاری، اگر گزینه‌های searchOperators و stringResult به true تنظیم نشوند، حالت toolbar search فوق را شاهد خواهیم بود. در غیراینصورت به حالت multi-field search سوئیچ می‌شود.
- در حالت جستجوی تک فیلدی توسط صفحه دیالوگ جستجوی jqGrid، فیلد در حال جستجو توسط searchField و مقدار آن توسط searchString به سرور ارسال شده‌اند. مابقی پارامترها نال هستند.
- در حالت جستجوی چند فیلدی توسط صفحه دیالوگ جستجوی jqGrid، اینبار filters مقدار دهی شده‌است و سایر پارامترها نال هستند. مقدار filters ارسالی، در حقیقت یک شیء JSON است با ساختار کلی ذیل:

```
{ "groupOp": "AND",
               "groups" : [
                 { "groupOp": "OR",
                     "rules": [
                         { "field": "name", "op": "eq", "data": "England" },
                         { "field": "id", "op": "le", "data": "5"}
                      ]
                 }
               ],
               "rules": [
                 { "field": "name", "op": "eq", "data": "Romania" },
                 { "field": "id", "op": "le", "data": "1"}
               ]
        }
```

که می‌توان چنین ساختاری را برای آن متصور شد:

```
public class SearchFilter
    {
        public string groupOp { set; get; }
        public List<SearchGroup> groups { set; get; }
        public List<SearchRule> rules { set; get; }
    }

    public class SearchRule
    {
        public string field { set; get; }
        public string op { set; get; }
        public string data { set; get; }

        public override string ToString()
        {
            return string.Format("'{0}' {1} '{2}'", field, op, data);
        }
    }

    public class SearchGroup
    {
        public string groupOp { set; get; }
        public List<SearchRule> rules { set; get; }
    }
```

در اینجا AND و Or کلی مشخص می‌شود، به همراه فیلدهای ارسالی به سرور، عملگرهای اعمالی بر روی آن‌ها و مقادیر مرتبط.
اگر این موارد را کنار هم قرار دهیم، به متدی عمومی ApplyFilter با امضای ذیل خواهیم رسید.

```
public IQueryable<T> ApplyFilter<T>(IQueryable<T> query, bool _search, string searchField, string
searchString,
string searchOper, string filters, NameValueCollection form)
```

کدهای کامل آن‌را به علت طولانی بودن پردازش سه حالت ذکر شده‌ی فوق، از پروژه‌ی پیوست می‌توانید دریافت کنید.
پس از آن، تغییراتی که در کدهای متد GetProducts باید اعمال شوند به صورت ذیل است:

```
[HttpPost]
        public ActionResult GetProducts(string sidx, string sord, int page, int rows,
                                        bool _search, string searchField, string searchString,
                                        string searchOper, string filters)
        {
            var list = ProductDataSource.LatestProducts;

            var pageIndex = page - 1;
            var pageSize = rows;
            var totalRecords = list.Count;
            var totalPages = (int)Math.Ceiling(totalRecords / (float)pageSize);

            var productsQuery = list.AsQueryable();

            productsQuery = new JqGridSearch().ApplyFilter(productsQuery, _search, searchField,
searchString,
                                                           searchOper, filters, this.Request.Form);
            var productsList = productsQuery.OrderBy(sidx + " " + sord)
                                            .Skip(pageIndex * pageSize)
                                            .Take(pageSize)
                                            .ToList();

            var productsData = new JqGridData
            {
                Total = totalPages,
                Page = page,
                Records = totalRecords,
                Rows = (productsList.Select(product => new JqGridRowData
                {
                    Id = product.Id,
                    RowCells = new List<string>
                                {
                                      product.Id.ToString(CultureInfo.InvariantCulture),
                                      product.Name,
                                      product.Supplier.CompanyName,
                                      product.Category.Name,
                                      product.Price.ToString(CultureInfo.InvariantCulture)
                                 }
                })).ToArray()
            };
            return Json(productsData, JsonRequestBehavior.AllowGet);
        }
```

- ابتدا چند پارامتر اضافه‌تر به امضای متد اضافه شده‌اند، تا فیلدهای جستجو را نیز دریافت کنند.
- نوع متد به HttpPost تغییر کرده‌است. این مورد برای ارسال اطلاعات حجیم جستجوها به سرور ضروری است و بهتر است از حالت Get استفاده نشود.
این حالت در سمت کلاینت نیز باید تنظیم شود:

```
$('#list').jqGrid({
//url access method type
mtype: 'POST',
```

- سایر سطرها مانند قبل است؛ فقط یک سطر ذیل جهت اعمال where پویا به عبارت LINQ ساخته شده، اضافه شده‌است:

```
productsQuery = new JqGridSearch().ApplyFilter(productsQuery, _search, searchField, searchString,
  searchOper, filters, this.Request.Form);
```

برای مطالعه بیشتر

جستجوی تک فیلدی

جستجوی نوار ابزاری

جستجوی چند فیلدی

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

jqGrid03.zip

نظرات خوانندگان

نویسنده: شهربانو مشهدی
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۵:۲

سلام؛ در حالت جستجو نوار ابزاری با زدن دوباره دکمه جستجو نوار ابزاری ، نوار ابزار حذف نمی‌شود. چه طور می‌شه با زدن دوباره دکمه نوار ابزار ، نوار ابزار از روی صفحه حذف شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۵:۲۴

با استفاده از متد toggleToolbar می‌شود نوار ابزار نمایش داده شده را مخفی یا مجددا نمایش داد:

```
<button onclick="$('#list')[0].toggleToolbar()">نمایش/حذف تولبار</button>
```

نویسنده: شهربانو مشهدی
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۷:۴۴

با سپاس؛ من کدتون رو به شکل زیر تبدیل کردم ولی برای بار اول باید دوبار بر روی دکمه نوار ابزار جستجو کلیک کرد آیا راه بهتری هم هست که همان بار اول درست کار کند؟ خیلی مچکرم

```
.jqGrid('navButtonAdd', "#pager", {
                caption: "نوار ابزار جستجو", title: "Search Toolbar", buttonicon: 'ui-icon-search',
                onClickButton: function () {
                    toolbarSearching();
                    $('#list')[0].toggleToolbar();
                }
            });
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۷:۵۳

- در کدهای شما ابتدا نوار ابزار نمایش داده می‌شود، سپس toggle تا خاموش شود.
+ این‌ها یک سری مثال هستند برای نمایش نحوه‌ی فعال سازی جداگانه‌ی این قابلیت‌ها از هم.
اگر می‌خواهید از ابتدای کار نوار ابزار جستجو نمایش داده شود، متد مربوطه را در انتهای کدهای jqGrid ذکر کنید:

```
$("#list").jqGrid({
        //...
    }).filterToolbar(options);
```

متدهای دیگر را هم به همین نحو «زنجیر وار» می‌توان ذکر کرد.
- سورس این گرید در فایل jquery.jqGrid.src.js قابل بررسی است. toggleToolbar را در آن جستجو کنید و از کدهای آن جهت یافتن tr.ui-search-toolbar و مخفی یا آشکار کردن آن ایده بگیرید.

نویسنده: شهربانو مشهدی
تاریخ: ۱۳۹۳/۰۴/۲۷ ۲۰:۱۰

با سپاس؛ خوشبختانه کد رو به شکل زیر تغییر دادم و نتیجه گرفتم:

```
$("#list").jqGrid({
        //...
    }).jqGrid('filterToolbar', { stringResult: true, searchOnEnter: true, autosearch: true,
searchOperators: true, groupOp: 'OR', defaultSearch: 'cn' })
            .jqGrid('navButtonAdd', "#pager", {
                caption: "نوار ابزار جستجو", title: "Search Toolbar", buttonicon: 'ui-icon-search',
                onClickButton: function () {
                    $("#list")[0].toggleToolbar();
```

```
            }
        });
        $("#list")[0].toggleToolbar();
```

نویسنده: محسن تقی پور
تاریخ: ۱۳۹۳/۰۵/۱۱ ۲:۳۷

با سلام؛ موقع لیست کردن دسته بندی‌ها در هنگام جستجو (در کمبو باکس) به ازای هر رکورد یک دسته بندی نمایش داده میشه از این کد استفاده کردم ولی نشد چکار باید بکنم تا درست نشون بده ؟

```
public ActionResult SuppliersSelect()
        {
            var list = BlNews.Select().Distinct().ToList();
            var suppliers = list.Select(x => new SelectListItem
            {
                Text = x.Admin.UserName,
                Value = x.Admin.Code.ToString(CultureInfo.InvariantCulture)
            }).ToList();
            return PartialView("_SelectPartial", suppliers);
        }
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۱۱ ۹:۳۴

« پیدا کردن آیتم‌های تکراری در یک لیست به کمک LINQ »