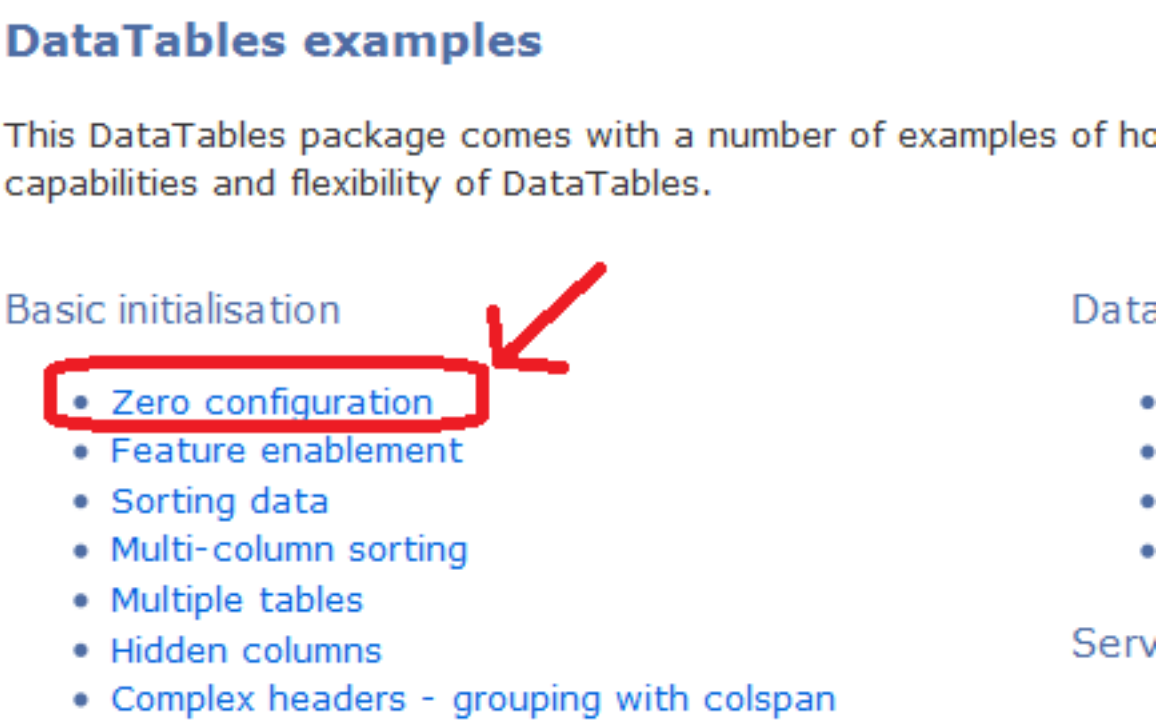
عنوان: **پلاگین DataTables کتابخانه jQuery - قسمت اول**
نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۱/۱۲/۲۹ ۱۲:۰
آدرس: www.dotnettips.info
برچسب‌ها: jQuery, DataTables

DataTables پلاگینی برای کتابخانه jQuery است. این پلاگین امکانات پیشرفته ای برای یک جدول html که حاوی داده‌ها است اضافه می‌کند، و همچنین عملیات صفحه بندی، جستجو، مرتب سازی داده‌ها را در سمت کاربر انجام می‌دهد.

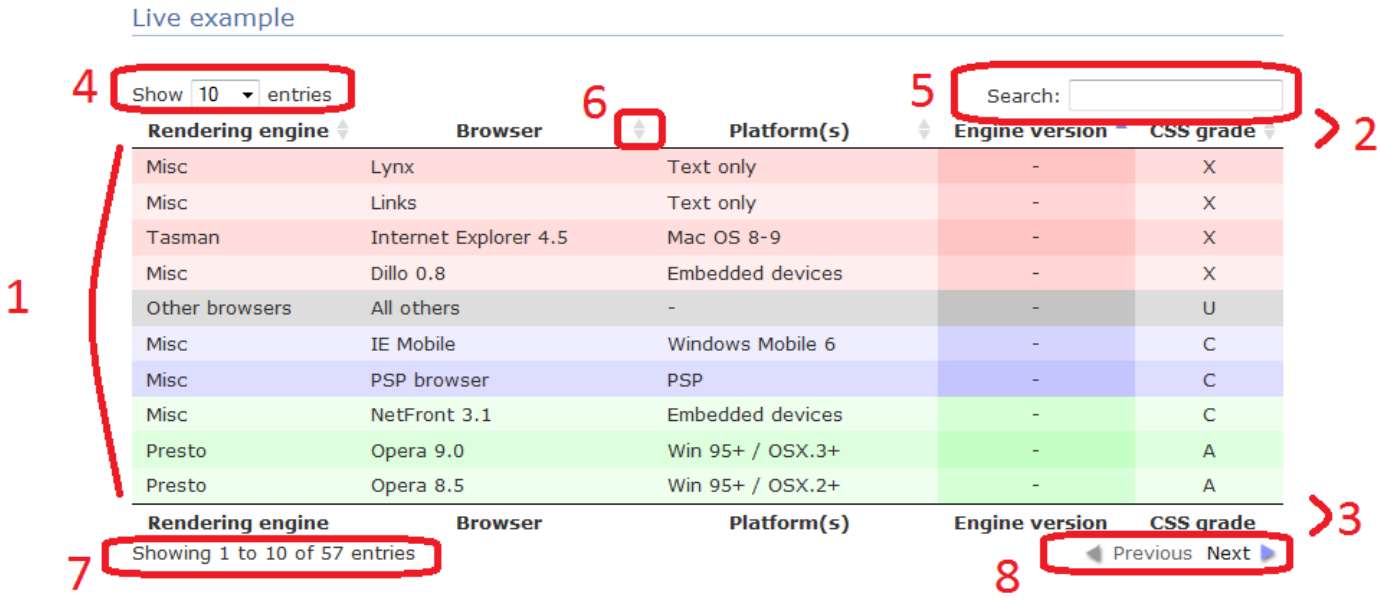
به طور خلاصه می‌توانید امکانات متعدد این پلاگین را در زیر مشاهده کنید:
صفحه بندی داده‌ها با تعداد رکوردهای قابل تغییر در هر صفحه (variable length pagination)
فیلتر کردن داده‌های بایند شده به جدول (on-the-fly filtering)

مرتب سازی داده‌ها بر اساس ستون‌های مختلف با قابلیت تشخیص نوع داده ستون (Multi-column sorting with data type detection)
تغییر اندازه ستون‌ها به صورت هوشمند (Smart handling of column widths)
نمایش داده‌ها در جدول از اکثر data sourceها (DOM، یک آرایه جاوا اسکریپتی، یک فایل، یا با استفاده از پردازش سمت سروری (سی شارپ، php و غیره))
قابلیت جهانی شدن یا منطبق شدن با زبان‌های مختلف دنیا (Fully Internationalisable)
قابلیت تعویض theme آن با استفاده از jQuery UI ThemeRoller
وجود داشتن 2900 آزمون واحد برای آن (backed by a suite of 2900 unit test)
وجود داشتن پلاگین‌های متعدد برای آن
و رایگان بودن آن
در این مقاله شما را به طور مقدماتی با این پلاگین آشنا خواهم کرد.
برای استفاده از این پلاگین ابتدا به اینجا مراجعه کرده و آنرا به همراه مثالهای آن که در یک فایل فشرده هستند را دانلود کنید.
بعد از دانلود و خارج کردن فایل دانلودی از حالت فشرده، وارد پوشه examples از آن که بشوید می‌توانید مثالهای متعدد در رابطه با این پلاگین را مشاهده نمائید.

مثال‌های این پلاگین یکی از بهترین منابع یادگیری آن هستند. در این سری از مقالات هم از روی همین مثالها پیش میرویم. برای این کار، بعد از مراجعه به پوشه examples فایل index.html را باز کنید و مثال اول را (Zero Configuration) کلیک کنید.



نتیجه حاصل از اجرای مثال Zero Configuration چیزی شبیه تصویر زیر است:



تصویر را شماره گزاری کرده ام تا بتوانم راحت‌تر آنرا برایتان تشریح کنم.

داده‌های درون جدول (10 تای اول) که در قسمت tbody جدول قرار دارند

قسمت thead جدول

قسمت tfoot جدول

اندازه صفحه (page size)

کادر جستجو که در کلیه ستون‌های جدول جستجویی را انجام می‌دهد و داده‌ها بر اساس آن فیلتر می‌شوند.

قابلیت مرتب سازی رکوردها بر اساس یک ستون خاص به صورت صعودی یا نزولی
اطلاعات مربوط به رکوردهای جاری و تعداد کل رکوردها
قابلیت تغییر صفحه با دکمه‌های previous و next

تشریح مثال Zero Configuration :

برای استفاده از این پلاگین، باید ارجاعی به کتابخانه jquery و نیز فایل jquery.dataTables.js وجود داشته باشد. این دو فایل در زیر پوشه media/js قرار گرفته اند.

```
<script type="text/javascript" language="javascript" src="../../media/js/jquery.js"></script>
<script type="text/javascript" language="javascript"
src="../../media/js/jquery.dataTables.js"></script>
```

و همچنین cssهای مربوطه به این پلاگین بدین صورت معرفی شده اند:

```
<style type="text/css" title="currentStyle">
    @import "../../media/css/demo_page.css";
    @import "../../media/css/demo_table.css";
</style>
```

در این مثال که ساده‌ترین مثال مربوط به این پلاگین است داده‌ها به صورت دستی در جدول قرار گرفته اند و روش‌های دیگر را به قسمت‌های بعد موکول می‌کنیم. اگر به source این مثال مراجعه کنید (از روی فایل اصلی و نه از طریق مرورگر) مشاهده می‌کنید که یک جدول html با id برابر با example وجود دارد که حاوی 57 سطر است (در قسمت tbody) که حاوی داده‌های جدول هستند. اما با مراجعه به source مثال از طریق مرورگر مشاهده می‌کنید تعداد این سطرها 10 تا هست و این بدین معنیه که پلاگین فقط تعداد رکوردهای مورد نیاز رو در قسمت tbody قرار می‌ده و از بقیه فاکتور می‌گیره و هر بار که کاربر به صفحه رو با دکمه‌های Previous و Next تغییر می‌ده این پلاگین قسمت tbody رو تغییر میده

این نکته هم جا نمونه که برای اعمال شدن پلاگین DataTables به یک جدول که به طور مثال id جدول example هست، به صورت زیر عمل می‌کنیم:

```
$(document).ready(function() {
    $('#example').dataTable();
} );
```

نظرات خوانندگان

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۱/۰۵ ۱۴:۰

سلام مطلب جالبی بود
اما یک س.ال برای من پیش اومده، آیا این پلاگین قادر است داده‌های جدول را از سمت سرور فراخوانی کند؟ یعنی می‌تواند با تغییر صفحه داده‌های صفحه‌های بعدی یا قبلی را از سرور (از دیتابیس) فراخوانی کند؟

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۱/۰۷ ۱۸:۴

سلام، بله این امکان هست که با تعویض صفحه توسط کاربر درخواستی به سرور فرستاده بشه و داده‌ها از سرور دریافت بشن.
در قسمتهای بعدی به آنها اشاره خواهم کرد

نویسنده: کامی
تاریخ: ۱۳۹۲/۰۵/۲۶ ۰:۲۴

سلام
شما نظرتون در مورد پلاگین jTable چیه؟ ایا امکانات jTable بیشتری از dataTable نیست؟

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۵/۲۶ ۲۱:۵۷

سلام
بنده با jTable کار نکردم ، نمی‌دونم امکاناتش چی هست . البته تعصب خاصی روی datatable ندارم. اگه شما اطلاعاتی دارید خوشحال می‌شیم اونو با دیگران به اشتراک بزارید. :)

نویسنده: وحید م
تاریخ: ۱۳۹۲/۰۹/۰۷ ۹:۳۲

با سلام؛ به‌نظر شما چه کتابخانه ای برای گرید مناسب ،کم حجم و حرفه ای‌تر ؟
گرید توکارmvc یا- DataTable-Jqgrid-Kendo-telerik-awesome خیلی مهمه ممنون از شما.

عنوان: **پلاگین DataTables کتابخانه jQuery - قسمت دوم**
نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۲/۰۶ ۰:۲۵
آدرس: www.dotnettips.info
برچسب‌ها: jQuery, DataTables

در قسمت قبلی شما را با DataTables آشنا کردم. به طور خلاصه نحوه اعمال کردن DataTables به یک جدول ساده html را گفتیم که با این کار به صورت پیش فرض، امکاناتی مثل فیلتر کردن داده ها، صفحه بندی و مرتب سازی آنها و نیز اعمال شدن استایل‌های css به همین جدول html خام اضافه می‌شود. نکته مهم در مثال قبلی این بود که داده‌های درون این جدول با کدنویسی خام html فراهم شدند، اما این را در نظر داشته باشید که اکثریت مواقع باید داده‌ها از یک بانک اطلاعاتی دریافت شوند و سپس درون جدول قرار بگیرند.

در این قسمت سعی خواهیم کرد تا منبع داده جدول را یک آرایه جاوا اسکریپتی و سپس کالکشنی از آبجکتهای جاوا اسکریپتی (json) در نظر بگیریم و نیز برخی ویژگی‌های پیش فرض پلاگین را غیر فعال نمائیم.

فرض کنید می‌خواهید لیستی از اطلاعات دانشجویان شامل نام (FirstName)، نام خانوادگی (LastName)، و سن (Age) را نمایش دهید. اطلاعات قرار است در جدول زیر قرار بگیرند:

```
<table id="std-grid">
      <thead>
           <th>نام</th>
           <th>نام خانوادگی</th>
           <th>سن</th>
     </thead>
     <tbody>

     </tbody>
</table>
```

مشاهده می‌کنید که این جدول فقط شامل قسمت header است و در بدنه آن هیچ سطری قرار نگرفته است. در این مثال اطلاعات از یک آرایه جاوا اسکریپتی باید خوانده شوند و تبدیل به html شده و در نهایت درون قسمت <tbody></tbody> آن تزریق شوند. خوشبختانه DataTables برای این کار امکانات آماده ای را در اختیار قرار می‌دهد. این کار بدین صورت قابل انجام است:

```
<script>
        $(document).ready(function() {
            $('#std-grid').dataTable({
                "aaData": [
                        ["24" ,"پارسائی" ,"پژمان"],
                        ["25" ,"الیاسی" ,"سعید"],
                        ["20" ,"گلزار" ,"محمد رضا"],
                        ["19" ,"ایرانی" ,"آرش"],
                        ["22" ,"فرمانی" ,"مرتضی"],
                        ["23" ,"حمیدیان" ,"سعید"],
                        ["23" ,"پارسانیا" ,"امین"],
                        ["24" ,"فقیهی" ,"محمد امین"],
                        ["25" ,"خرمی" ,"محمد"],
                        ["20" ,"امیریان" ,"سینا"],
                        ["19" ,"ایرانی" ,"آرش"],
                        ["22" ,"فرزانه" ,"وحید"],
                        ["23" ,"فرمانی" ,"امیر علی"],
                        ["23" ,"حسینی" ,"امین"],
            });
        });

    </script>
```

شرح کد:
aaData : یک آرایه دو بعدی (که به آن ماتریس یا آرایه ای از آرایه‌ها هم گفته می‌شود) است که مقادیر سلول هایی را نشان

می‌دهد که در جدول قرار خواهند گرفت. تعداد ستون‌ها در این آرایه دو بعدی باید با تعداد ستون‌های جدول html متناظر یکسان باشند.

در مثال بالا از یک ماتریس به عنوان منبع داده استفاده شد. منبع داده می‌تواند به فرمت json نیز باشد. البته در این صورت باید ستون‌های جدول html را هم به پلاگین معرفی کنید، بدین صورت:

```
$(document).ready(function() {
            $('#std-grid').dataTable({
                "aaData": [
                    {"FirstName" : "پژمان", "LastName" : "پارسائی", "Age" : "24"},
                    { "FirstName": "سعید", "LastName": "الیاسی", "Age": "25" },
                    { "FirstName": "محمد رضا", "LastName": "گلزار", "Age": "24" },
                    { "FirstName": "آرش", "LastName": "ایرانی", "Age": "24" },
                    { "FirstName": "مرتضی", "LastName": "فرمانی", "Age": "24" },
                    { "FirstName": "سعید", "LastName": "حمیدیان", "Age": "24" },
                    { "FirstName": "امین", "LastName": "پارسانیا", "Age": "24" },
                    { "FirstName": "محمد امین", "LastName": "فقیهی", "Age": "24" },
                    { "FirstName": "محمد", "LastName": "خرمی", "Age": "24" },
                    { "FirstName": "سینا", "LastName": "امیریان", "Age": "24" },
                    { "FirstName": "آرش", "LastName": "ایرانی", "Age": "24" },
                    { "FirstName": "وحید", "LastName": "فرزانه", "Age": "24" },
                    { "FirstName": "امیر علی", "LastName": "فرمانی", "Age": "24" },
                    { "FirstName": "امین", "LastName": "حسینی", "Age": "24" },
                ],
                "aoColumns": [
                      { "mDataProp": "FirstName" },
                      { "mDataProp": "LastName" },
                      { "mDataProp": "Age" }
                ]
            });
        });
```

aaData : همان طور که گفته شد در این قسمت دیتاهای درون جدول آورده می‌شوند و در این مثال آنها به فرمت json نوشته شده اند.
aoColumns : در این قسمت باید اسم ستون‌های جدول ذکر شوند.

غیرفعال کردن بعضی از ویژگی‌های پیش فرض DataTables

همان طور که گفته شد پلاگین DataTables به صورت پیش فرض ویژگی‌های مرتب سازی (sorting)، صفحه بندی (paging)، فیلتر کردن داده‌ها (filtering)، و غیره را به جدول مورد نظرش اعمال می‌کند. و بدین صورت قابل تغییر است:

```
$('#std-grid').dataTable({
          "bPaginate": false,
          "bLengthChange": false,
          "bFilter": false,
          "bSort": false,
          "bInfo": true,
          "bAutoWidth": false
});
```

bPaginate : بیان می‌کند آیا صفحه بندی سطرهای جدول فعال باشد یا نه.
bLengthChange : در صورتی که قابلیت صفحه بندی فعال باشد ، بیان می‌کند که کاربر بتواند اندازه صفحه را تغییر دهد یا نه.
bFilter : بیان می‌کند آیا قابلیت فیلتر کردن داده‌ها فعال باشد یا نه.
bSort : بیان می‌کند قابلیت مرتب سازی داده‌های جدول فعال باشد یا نه.
bInfo : بیان می‌کند که قسمت info زیر گرید نشان داده شود یا نه (در این قسمت اطلاعاتی راجع به تعداد کل رکوردهای یابند شده به جدول و نیز رکوردهای درون صفحه جاری نشان داده می‌شود)
bAutoWidth : در صورتی که این گزینه فعال باشد اندازه عرض هر ستون به صوتر خودکار توسط DataTables مقدار دهی خواهد شد.

مقدارهای قابل قبول برای هر کدام از این خصوصیات : true یا false

کدهای مربوط به این مثال را می‌توانید از لینک زیر دریافت کنید:
DataTables-DoteNetTips-Tutorial-02.zip

نظرات خوانندگان

نویسنده: farzad
تاریخ: ۱۳۹۲/۰۴/۲۵ ۲۰:۵

با سلام
آیا این پلاگین قابلیت حذف و یا ویرایش داده‌ها رو هم می‌ده؟

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۴/۲۵ ۲۱:۲۷

سلام
بله، نمونه پیاده سازی شده در mvc رو می‌تونید توی لینک زیر مشاهده کنید:

(ASP.NET MVC Editable Table (jQuery DataTables and ASP.NET MVC integration - Part II

عنوان: **پلاگین DataTables کتابخانه jQuery - قسمت سوم**
نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۲/۲۰ ۲۱:۲۵
آدرس: www.dotnettips.info
برچسب‌ها: jQuery, DataTables, jQuery-Plugins

در این قسمت اطلاعات را به صورت ajax از یک فایل متنی می‌خوانیم و آنها را در جدول قرار می‌دهیم. سپس به سفارشی کردن بعضی از قسمت‌های DataTables خواهیم پرداخت.

دریافت اطلاعات به صورت ajax از یک فایل متنی

فرض کنید که اطلاعات در یک فایل txt به صورت اشیاء جاوا اسکریپتی ذخیره شده اند، و این فایل بر روی سرور قرار دارد. می‌خواهیم از این فایل به عنوان منبع داده استفاده کرده و اطلاعات درون آن را به صورت ajax دریافت کرده و در یک جدول html تزریق کنیم. خوشبختانه با استفاده از امکاناتی که این پلاگین تهیه کرده است این کار به سادگی امکان پذیر است.

همان طور که در اینجا بیان شده است ، فرض کنید که جدولی داشته باشیم و بخواهیم اطلاعات راجع به مرورگرهای مختلف را در آن نمایش دهیم. قصد داریم این جدول شامل قسمتهای header و footer و نیز body باشد، بدین صورت:

```
<table id="browsers-grid">
    <thead>
       <tr>
          <th width="20%">موتور رندرگیری</th>
          <th width="25%">مرورگر</th>
          <th width="25%">پلتفرم (ها)</th>
          <th width="15%">نسخه موتور</th>
          <th width="15%">نمره css</th>
       </tr>
    </thead>

    <tbody>

    </tbody>

    <tfoot>
       <tr>
          <th>موتور رندرگیری</th>
          <th>مرورگر</th>
           <th>پلتفرم (ها)</th>
          <th>نسخه موتور</th>
          <th>نمره css</th>
       </tr>
    </tfoot>
</table>
```

برای هر ستون از این جدول عرضی در نظر گرفته شده است. اگر این کار انجام نشود به صورت خودکار به تمام ستونها عرض داده می‌شود.

داده هایی که باید در بدنه جدول قرار بگیرند، در یک فایل متنی روی سرور قرار دارند. محتویات این فایل چیزی شبیه زیر است:

```
{
   "aaData": [
      {"engine":"Trident", "browser":"Internet Explorer 4.0", "platform":"Win95+", "version":"4",
"grade":"X"},
      {"engine":"Trident", "browser":"Internet Explorer 5.0", "platform":"Win95+", "version":"5",
"grade":"C"},
      {"engine":"Trident", "browser":"Internet Explorer 5.5", "platform":"Win95+", "version":"5.5",
"grade":"A"}
   ]
}
```

همان طور که مشاهده می‌کنید فرمت ذخیره داده‌ها در این فایل به صورت json یا اشیاء جاوا اسکریپتی است. این اشیاء باید به خصوصیت aaData نسبت داده شوند که در قسمت قبل راجع به آن توضیح دادیم. تعداد این اشیاء 57 تا بود که برای سادگی بیشتر 3 تا از آنها را اینجا ذکر کردیم.

اسکریپتی که داده‌ها را از فایل متنی خوانده و آنها را در جدول قرار می‌دهد هم بدین صورت خواهد بود:

```
$(document).ready(function () {
        $('#browsers-grid').dataTable({
            "sAjaxSource": "datasource/objects.txt",
            "bProcessing": true,
            "aoColumns": [
                { "mDataProp": "engine" },
                { "mDataProp": "browser" },
                { "mDataProp": "platform" },
                { "mDataProp": "version" },
                { "mDataProp": "grade" }
            ]
    });
});
```

شرح کد:

sAjaxSource : رشته

نوع داده ای که قبول می‌کند رشته ای و بیان کننده آدرسی است که داده‌ها باید از آنجا دریافت شوند. در اینجا داده‌ها در فایل متنی objects.txt در پوشه datasource قرار دارند.

bProcessing : بولین

نوع داده‌های قابل قبول این خصوصیت true یا false هست و بیان کننده این است که یک پیغام loading تا زمانی که داده‌ها دریافت شوند و در جدول قرار بگیرند نمایش داده شوند یا خیر.

تنظیم کردن گزینه‌های اضافی دیگر

sAjaxDataProp : رشته

همان طور که گفتیم در فایل متنی که حاوی اشیاء json بود ، این اشیاء را به متغیری به اسم aaData منتسب کردیم. این نام را می‌توان تغییر داد مثلا فرض کنید در فایل متنی داده‌ها به متغیری به اسم data منتسب شده اند:

```
{
   "data": [
      {"engine":"Trident", "browser":"Internet Explorer 4.0", "platform":"Win95+", "version":"4",
"grade":"X"},
      {"engine":"Trident", "browser":"Internet Explorer 5.0", "platform":"Win95+", "version":"5",
"grade":"C"},
      {"engine":"Trident", "browser":"Internet Explorer 5.5", "platform":"Win95+", "version":"5.5",
"grade":"A"}
   ]
}
```

در این صورت باید خصوصیت sAjaxDataProp را به همان نامی که در فایل متنی مشخص کرده اید مقداردهی کنید، در غیر این صورت داده‌های جدول هیچ گاه بارگذاری نخواهند شد. بدین صورت:

```
"sAjaxDataProp": "data"
```

یا اگر داده‌ها را بدین صورت در فایل متنی ذخیره کرده اید:

```
{ "data": { "inner": [...] } }
```

آنگاه خصوصیت sAjaxDataProp بدین صورت مقداردهی خواهد شد:

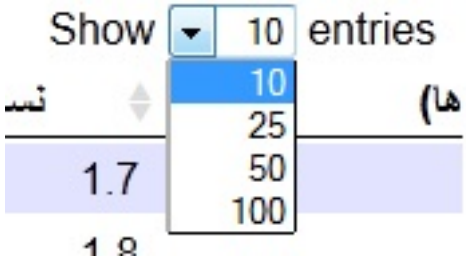
```
"sAjaxDataProp": "data.inner"
```

sPaginationType : رشته
نحوه صفحه بندی و حرکت بین صفحات مختلف را بیان می‌کند. اگر با two_buttonمقدار دهی شود (مقدار پیش فرض) حرکت بین صفحات مختلف به وسیله دکمه‌های Next و Previous امکان پذیر خواهد بود. اگر با full_numbersمقدار دهی شود حرکت بین صفحات با دکمه‌های Next و Previous ، و همچنین دکمه‌های First و Last و نیز شماره صفحه فعلی و دو صفحه بعدی و دو صفحه قبلی قابل انجام است.

شکل الف) صفحه بندی به صورت full_numbers

bLengthChange : بولین
بیان می‌کند کاربر بتواند اندازه صفحه را تغییر دهید یا نه. به صورت پیش فرض این گزینه true است. اگر آن به false مقدار دهی شود لیست بازشونده مربوط به اندازه صفحه مخفی خواهد شد.

aLengthMenu : آرایه یک بعدی یا دو بعدی
به صورت پیش فرض در لیست باز شونده مربوط به تعداد رکوردهای قابل نمایش در هر صفحه اعداد 10 ، 25 ، 50 ، و 100 قرار دارند.



شکل ب) لیست بازشونده شامل اندازه‌های صفحه

در صورتی که بخواهیم این گزینه‌ها را تغییر دهیم باید خصوصیت aLengthMenu را مقدار دهی کنیم. اگر مقداری که به این خصوصیت می‌دهیم یک آرایه یک بعدی باشد، مثلا

```
"aLengthMenu": [25, 50, 100, -1],
```

نتیجه یک لیست باز شوند است که دارای چهار عنصر است که value و text آنها یکی است. (نکته: چهارمین عنصر از لیست بالا دارای مقدار 1- خواهد بود که با انتخاب این گزینه تمام رکوردها نمایش می‌یابند). اما اگر می‌خواهیم که text و value این عناصر با هم فرق کند از یک آرایه دو بعدی استفاده خواهیم کرد، مثلا:

```
"aLengthMenu": [[25, 50, 100, -1], ["بیست و پنج", "پنجاه", "صد", "همه"]],
```

iDisplayLength : عدد صحیح
تعداد رکوردهای قابل نمایش در هر صفحه هنگامی که داده‌ها در جدول ریخته می‌شوند را معین می‌کند. می‌توانید این را مقداری بدهید که در خصوصیت aLengthMenu ذکر نشده است، مثلا 28 تا.

sDom : رشته
پلاگین DataTables به صورت پیش فرض لیست بازشونده اندازه صفحه و کادر متن مربوط به جستجو را در بالای جدول داده‌ها اضافه می‌کند، و نیز اطلاعات دیگر و همچنین امکانات مربوط به صفحه بندی را به قسمت پایین جدول اضافه می‌کند. شما می‌توانید موقعیت این عناصر را با استفاده از پارامتر sDom تغییر دهید.

نحو (syntax) مقداری که پارامتر sDom قبول می‌کند مقداری عجیب و غریب است، مثلا:

```
'<"top"iflp<"clear">>rt<"bottom"iflp<"clear">>'
```

این خط بیان می‌کند که در قسمت بالای جدول یک تگ div با کلاس top قرار بگیرد. در این تگ قسمت اطلاعات (یعنی Showing x to xx from xxx entries) (با حرف i) ، کادر جستجو (با حرف f) ، لیست بازشونده مربوط به اندازه صفحه (با حرف l) ، و نیز قسمت صفحه بندی (با حرف p)قرار خواهند گرفت. در انتهای تگ div با کلاس top، یک تگ div با کلاس clear قرار خواهد گرفت. بعد قسمت مربوط به پیغام loading (با حرف r) و بعد با حرف t جدول حاوی داده‌ها قرار می‌گیرد. در نهایت یک تگ div با کلاس bottom قرار می‌گیرد و با حرفهای i ، و f ، و l و p درون آن قسمتهای اطلاعات ، کادرجستجو، لیست بازشونده اندازه صفحه و نیز قسمت صفحه بندی قرار خواهد گرفت و در نهایت یک تگ div با کلاس clear قرار خواهد گرفت.

حرفهایی که در sDom معنی خاصی می‌دهند :
l سر حرف Length Changing برای لیست بازشونده مربوط به اندازه صفحه
f سر حرف Filtering input برای قسمت کادر جستجو
t سرحرف table برای جدول حاوی داده ها
i سر حرف information برای قسمت Showing x to xx from xxx entries
p سر حرف pagination برای قسمت صفحه بندی
r حرف دوم pRocessing برای قسمت پیغام قبل از بار کردن داده‌های جدول (قسمت loading)
H و F که مربوط به theme‌های jQuery UI می‌شوند که بعدا درباره آنها توضیح داده می‌شود.

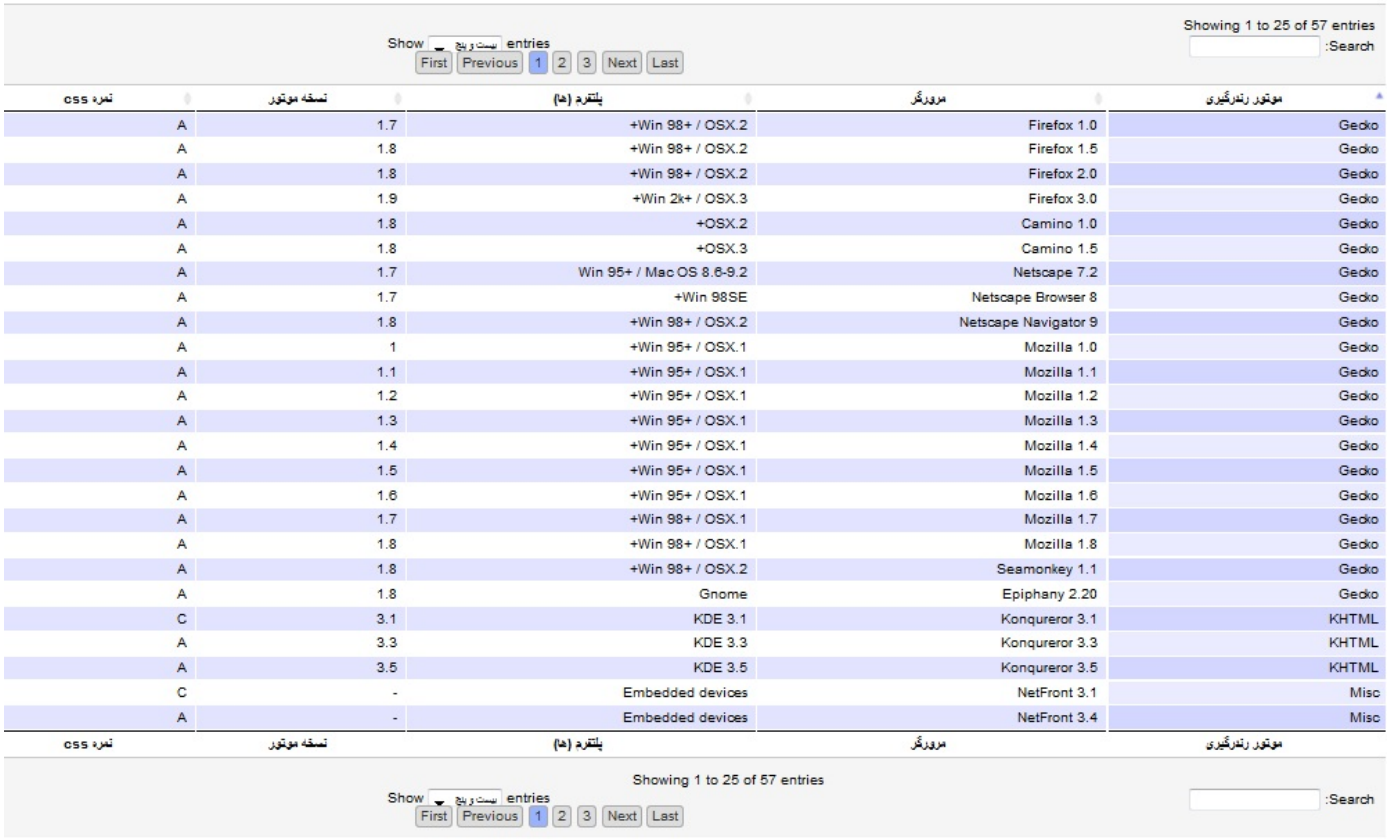
همچنین بین علامت‌های کوچکتر (<) و بزرگتر (>) یعنی اگر چیزی بیاید در یک تگ div قرار خواهد گرفت. اگر بخواهیم div ی بسازیم و به آن کلاس بدهیم از نحو زیر استفاده خواهیم کرد:

```
'<"class" and '>'
```

و اگر بخواهیم یک تگ div با یک id مشخص بسازیم از نحو زیر استفاده خواهیم کرد:

```
'<"#id" and '>'
```

در نهایت جدولی مثل جدول زیر تولید خواهد شد:



Showing 1 to 25 of 57 entries
Search:

Show بیست و پنج entries
First Previous 1 2 3 Next Last

موتور رندرگیری	مرورگر	پلتفرم (ها)	نسخه موتور	نمره css
Gecko	Firefox 1.0	+Win 98+ / OSX.2	1.7	A
Gecko	Firefox 1.5	+Win 98+ / OSX.2	1.8	A
Gecko	Firefox 2.0	+Win 98+ / OSX.2	1.8	A
Gecko	Firefox 3.0	+Win 2k+ / OSX.3	1.9	A
Gecko	Camino 1.0	+OSX.2	1.8	A
Gecko	Camino 1.5	+OSX.3	1.8	A
Gecko	Netscape 7.2	Win 95+ / Mac OS 8.6-9.2	1.7	A
Gecko	Netscape Browser 8	+Win 98SE	1.7	A
Gecko	Netscape Navigator 9	+Win 98+ / OSX.2	1.8	A
Gecko	Mozilla 1.0	+Win 95+ / OSX.1	1	A
Gecko	Mozilla 1.1	+Win 95+ / OSX.1	1.1	A
Gecko	Mozilla 1.2	+Win 95+ / OSX.1	1.2	A
Gecko	Mozilla 1.3	+Win 95+ / OSX.1	1.3	A
Gecko	Mozilla 1.4	+Win 95+ / OSX.1	1.4	A
Gecko	Mozilla 1.5	+Win 95+ / OSX.1	1.5	A
Gecko	Mozilla 1.6	+Win 95+ / OSX.1	1.6	A
Gecko	Mozilla 1.7	+Win 98+ / OSX.1	1.7	A
Gecko	Mozilla 1.8	+Win 98+ / OSX.1	1.8	A
Gecko	Seamonkey 1.1	+Win 98+ / OSX.2	1.8	A
Gecko	Epiphany 2.20	Gnome	1.8	A
KHTML	Konqureror 3.1	KDE 3.1	3.1	C
KHTML	Konqureror 3.3	KDE 3.3	3.3	A
KHTML	Konqureror 3.5	KDE 3.5	3.5	A
Misc	NetFront 3.1	Embedded devices	-	C
Misc	NetFront 3.4	Embedded devices	-	A

Showing 1 to 25 of 57 entries
Search:

Show بیست و پنج entries
First Previous 1 2 3 Next Last

شکل ج) جدول نهایی تولید شده توسط DataTables

کدهای نهایی این مثال را از DataTables-DoteNetTips-Tutorial-03.zip دریافت کنید.

نظرات خوانندگان

نویسنده: sorosh
تاریخ: ۱۳۹۲/۰۴/۰۷ ۷:۴۷

با سلام و عرض ادب
زمانیکه من با Jquery , Ajax سطرهای دیتای جدول مورد نظر برای DataTable شدن را از سمت سرور ایجاد می‌کنم متاسفانه بار اول دیتاها رو نشون میده ولی Search نمیکنه و صفحه بندی هم نمیکنه و ... در ضمن کدهای مربوطه رو هم می‌گذارم . لطفا راهنمایی کنید که اگه خواستیم دیتاها را از سمت سرور بیاریم و کار بده باید چه کار کرد؟ مرسی

```
$(document).ready(function () {
                dataparam2 = "cmd=FillScope";
                $.ajax({
                    url: "Default2.aspx",
                    type: "POST",
                    data: dataparam2,
                    async: true,
                    success: function (msg) {
                        if (msg != '') {
                            var data = eval("(" + msg + ")");
                            $("#tbodytblMain").html('');
                            for (var i = 0; i < data.length; i++) {
                                $("#tbodytblMain").append(
                                        "<tr class='odd gradeX'>"
                                        + "<td style='width:200px'>" + data[i].T + "</td>"
                                        + "<td style='width:150px'>" + data[i].P + "</td>"
                                         + "<td>" + data[i].S + "</td>"
                                        + "<td>" + data[i].TP + "</td>"
                                        + "<td>" + data[i].Sp + "</td>" + "</tr>");

                            }
                        }
                    },
                    error: function (msg) {

                    }
                });


                $('#tblMain').dataTable();
            });
```

کد سمت سرور:

```
if (Request["cmd"] == "FillScope")
        {
            string Val = "برخوار";
            JavaScriptSerializer js = new JavaScriptSerializer();
            string serText = "";
            MUIDataClassesDataContext db = new MUIDataClassesDataContext();
            var LST = (from x in db.tblProjectInfos
                       where x.tblScope.xScopeName.Contains(Val)
                       orderby x.tblScope.xScopeName
                       select new
                       {

                           P = x.xPlace,
                           S = x.tblScope.xScopeName,
                           TP = x.tblProjectType.xProjectTypeName,
                           Sp = x.tblStatus.xStatusName
                       });
            serText = js.Serialize(LST);
            Response.Write(serText);
            Response.End();

        }
```

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۴/۰۸ ۳:۱۱

سلام
رندر کردن جدول حاوی داده‌ها باید به data tables سپرده بشه. بدین صورت که داده‌های دریافتی از سرور به فرمت مناسبی تبدیل بشن و بعد به خصوصیت aaData نسبت داده بشن، البته به تبع اون و حتما باید خصوصیت aoColumns هم مقدار دهی بشه.

```
$(document).ready(function () {
      $.ajax({
          url: "Default.aspx/GetBrowsers",
          contentType: "application/json; charset=utf-8",
          dataType: "json",
          type: "POST",
          success: function (response) {
          if (response != "") {
                    var data = eval("(" + response.d + ")");
                    $('#browsers-grid').dataTable({
                            "aaData": data,
                            "bProcessing" : true,
                            "aoColumns": [
                                { "mData": "Engine" },
                                { "mData": "Name" },
                                { "mData": "Platform" },
                                { "mData": "Version", "sClass": "center" },
                                { "mData": "Grade", "sClass": "center" }
                            ]
                    });
              }
          },
      });
});
```

کدهای سمت سرور:
مثلا فرض کنید ذر سمت سرور بخواهید لیستی از مرورگرها رو برگشت بدین. کلاس زیر رو در نظر بگیرید:

```
public class Browser
{
    public int Id { get; set; }
    public string Engine { get; set; }
    public string Name { get; set; }
    public string Platform { get; set; }
    public float Version { get; set; }
    public string Grade { get; set; }
}
```

برای برگشت دادن لیستی از مرورگرها به طرف کلاینت، متدی مثل زیر خواهید داشت:

```
[WebMethod]
public static string GetBrowsers()
{
    List<Browser> browsers = new List<Browser>()
        {
            new Browser
                {
                    Id = 1,
                    Engine = "Trident",
                    Name = "Internet Explorer 4.0",
                    Platform = "Win95+",
                    Version = 4,
                    Grade = "X"
                },
            new Browser
                {
                    Id = 2,
                    Engine = "Trident",
                    Name = "Internet Explorer 5.0",
```

```
                    Platform = "Win95+",
                    Version = 5,
                    Grade = "C"
                },
        };
    return browsers.ToJson();
}
```

در متد بالا، لیستی از مرورگرها با استفاده از یک متد الحاقی تبدیل به فرمت json میشه و به طرف کاربر فرستاده میشه.

عنوان: **پلاگین DataTables کتابخانه jQuery - قسمت چهارم**
نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۴/۲۵ ۷:۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net MVC, jQuery, DataTables, jQuery-Plugins

همان طور که قبلا اشاره کردیم، این پلاگین می‌تواند از یک زبان برنامه نویسی سمت سرور داده‌های مورد نیاز خودش را دریافت کند. می‌توانید داده‌ها را با استفاده از AJAX و به صورت JSON از سرور دریافت کرده و با استفاده از DataTables آنها را در جدول تزریق کنید. در این قسمت سعی خواهیم کرد تا با استفاده از jQuery DataTables یک گرید را در MVC ایجاد کنیم. البته برای حذف جزئیات داده‌ها به جای این که از یک بانک اطلاعاتی دریافت شوند، در حافظه ساخته می‌شوند. در هر صورت اساس کار یکی است.

قصد داریم تا مانند مثال قسمت قبل، مجموعه ای از اطلاعات مربوط به مرورگرهای مختلف را در یک جدول نشان دهیم، اما این بار منبع داده ما فرق می‌کند. منبع داده از طرف سرور فراهم می‌شود. هر مرورگر - همان طور که در قسمت قبل مشاهده نمودید - شامل اطلاعات زیر خواهد بود:

موتور رندرگیری (Engine)

نام مرورگر (Name)

پلتفرم (Platform)

نسخه موتور (Version)

نمره سی اس اس (Grade)

به همین دلیل در سمت سرور، کلاسی خواهیم ساخت که نمایانگر یک مرورگر باشد. بدین صورت:

```
public class Browser
{
    public int Id { get; set; }
    public string Engine { get; set; }
    public string Name { get; set; }
    public string Platform { get; set; }
    public float Version { get; set; }
    public string Grade { get; set; }
}
```

استفاده از روش server side processing برای دریافت داده‌ها از سرور

این روش، یکی از امکانات jQuery DataTables است که با استفاده از آن، کلاینت تنها یک مصرف کننده صرف خواهد بود و وظیفه پردازش اطلاعات - یعنی تعداد رکوردهایی که برگشت داده می‌شود، صفحه بندی، مرتب سازی، جستجو، و غیره - به عهده سرور خواهد بود.

برای به کار گیری این روش، اولین کار این است که ویژگی bServerSide را true کنیم، مثلا بدین صورت:

```
var $table = $('#browsers-grid');
$table.dataTable({
      "bServerSide": true,
      "sAjaxSource": "/Home/GetBrowsers"
 });
```

همچنین ویژگی sAjaxSource را به Url ی که باید داده‌ها از آن دریافت شوند مقداردهی می‌کنیم.

به صورت پیش فرض مقدار ویژگی bServerSide مقدار false است؛ که یعنی منبع داده این پلاگین از سمت سرور خوانده نشود. اگر true باشد منبع داده و خیلی اطلاعات دیگر مربوط به داده‌های درون جدول باید از سرور به مرورگر کاربر پس فرستاده شوند. با true کردن مقدار bServerSide، آنگاه DataTables اطلاعاتی را راجع به شماره صفحه جاری، اندازه هر صفحه، شروط فیلتر کردن داده ها، مرتب سازی ستون ها، و غیره را به سرور می‌فرستد. همچنین انتظار می‌رود تا سرور در پاسخ به این درخواست، داده‌های مناسبی را به فرمت JSON به مرورگر پس بفرستد. در حالتی که bServerSide مقدار true به خود بگیرد، پلاگین فقط رابطه متقابل بین کاربر و سرور را مدیریت می‌کند و هیچ پردازشی را انجام نمی‌دهد.

در این درخواست XHR یا Ajax ی پارامترهایی که به سرور ارسال می‌شوند این‌ها هستند:

iDisplayStart عدد صحیح
نقطه شروع مجموعه داده جاری

iDisplayLength عدد صحیح
تعداد رکوردهایی که جدول می‌تواند نمایش دهد. تعداد رکوردهایی که از طرف سرور برگشت داده می‌شود باید با این عدد یکسان باشند.

iColumns عدد صحیح
تعداد ستونهایی که باید نمایش داده شوند.

sSearch رشته
فیلد جستجوی عمومی

bRegex بولین
اگر true باشد معنی آن این است که می‌توان از عبارات باقاعده برای جستجوی عبارتی خاص در کل ستون‌های جدول استفاده کرد. مثلا در کادر جستجو نوشت :

```
^[1-5]$
```

که یعنی 1 و 5 همه عددهای بین 1و 5.

bSearchable_(int) بولین
نمایش می‌دهد که یک ستون در طرف کاربر قابلیت searchable آن true هست یا نه.

sSearch_(int) رشته
فیلتر مخصوص هر ستون. اگر از ویژگی multi column filtering پلاگین استفاده شود به صورت ، sSearch1 ، sSearch0 sSeach2 و ... به طرف سرور ارسال می‌شوند. شماره انتهای هر کدام از پارامترها بیانگر شماره ستون جدول است.

bRegex_(int) بولین
اگر true باشد، بیان می‌کند که می‌توان از عبارت با قاعده در ستون شماره int جهت جستجو استفاده کرد.

bSortable_(int) بولین

مشخص می‌کند که آیا یک ستون در سمت کلاینت، قابلیت مرتب شدن بر اساس آن وجود دارد یا نه. (در اینجا int اندیس ستون را مشخص می‌کند)

iSortingCols عدد صحیح
تعداد ستون هایی که باید مرتب سازی بر اساس آنها صورت پذیرد. در صورتی که از امکان multi column sorting استفاده کنید این مقدار می‌تواند بیش از یکی باشد.

iSortCol_(int) عدد صحیح
شماره ستونی که باید بر اساس آن عملیات مرتب سازی صورت پذیرد.

sSortDir_(int) رشته
نحوه مرتب سازی ؛ شامل صعودی (asc) یا نزولی (desc)

mDataProp_(int) رشته
اسم ستون‌های درون جدول را مشخص می‌کند.

sEcho رشته
اطلاعاتی که datatables از آن برای رندر کردن جدول استفاده می‌کند.

شکل زیر نشان می‌دهد که چه پارامترهایی به سرور ارسال می‌شوند.

Parameters application/x-www-form-urlencoded

bRegex	false
bRegex_0	false
bRegex_1	false
bRegex_2	false
bRegex_3	false
bRegex_4	false
bSearchable_0	true
bSearchable_1	true
bSearchable_2	true
bSearchable_3	true
bSearchable_4	true
bSortable_0	true
bSortable_1	true
bSortable_2	true
bSortable_3	true
bSortable_4	true
iColumns	5
iDisplayLength	25
iDisplayStart	0
iSortCol_0	1
iSortingCols	1
mDataProp_0	Engine
mDataProp_1	Name
mDataProp_2	Platform
mDataProp_3	Version
mDataProp_4	Grade
sColumns	
sEcho	2
sSearch	
sSearch_0	
sSearch_1	
sSearch_2	
sSearch_3	
sSearch_4	
sSortDir_0	desc

شکل ب) پارامترهای ارسالی به سرور به صورت json

بعضی از این پارامترها بسته به تعداد ستون‌ها قابل تغییر هستند. (آن پارامترهایی که آخرشان یک عدد هست که نشان دهنده شماره ستون مورد نظر می‌باشد)

در پاسخ به هر درخواست XHR که datatables به سرور می‌فرستد، انتظار دارد تا سرور نیز یک شیء json را با فرمت مخصوص که شامل پارامترهای زیر می‌شود به او پس بفرستد:

iTotalRecords عدد صحیح
تعداد کل رکوردها (قبل از عملیات جستجو) یا به عبارت دیگر تعداد کل رکوردهای درون آن جدول از دیتابیس که داده‌ها باید از آن دریافت شوند. تعداد کل رکوردهایی که در طرف سرور وجود دارند. این مقدار فقط برای نمایش به کاربر برگشت داده می‌شود و نیز از آن برای صفحه بندی هم استفاده می‌شود.

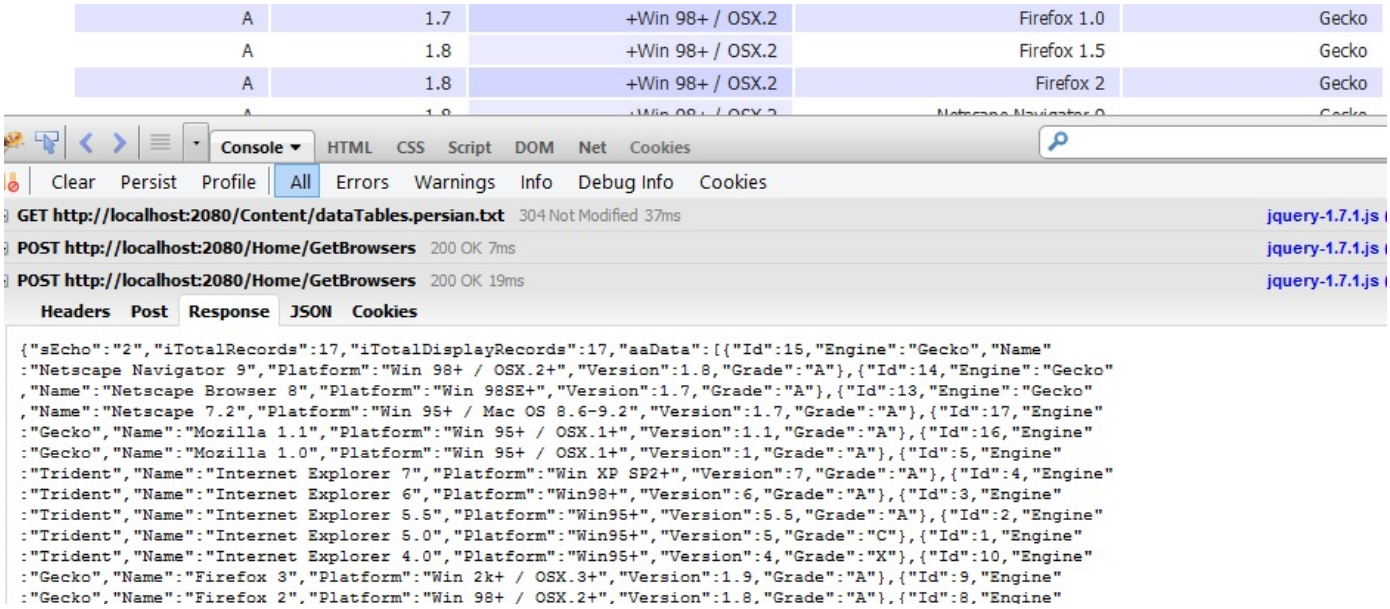
iTotalDisplayRecords عدد صحیح
تعداد کل رکوردها (بعد از عملیات جستجو) یا به عبارت دیگر تعداد کل رکوردهایی که بعد از عملیات جستجو پیدا می‌شوند نه فقط آن تعداد رکوردی که به کاربر پس فرستاده می‌شوند. تعداد کل رکوردهایی که با شرط جستجو مطابقت دارند. اگر کاربر چیزی را جستجو نکرده باشد مقدار این پارامتر با پارامتر iTotalRecords یکسان خواهد بود.

sEcho عدد صحیح
یک عدد صحیح است که در قالب رشته در تعامل بین سرور و کلاینت جا به جا می‌شود. این مقدار به ازاء هر درخواست تغییر می‌کند. همان مقداری که مرورگر به سرور می‌دهد را سرور هم باید به مرورگر تحویل بدهد. برای جلوگیری از حملات XSS باید آن را تبدیل به عدد صحیح کرد. پلاگین DataTables مقدار این پارامتر را برای هماهنگ کردن و منطبق کردن درخواست ارسال شده و جواب این درخواست استفاده می‌کند. همان مقداری که مروگر به سرور می‌دهد را باید سرور تحویل به مرورگر بدهد.

sColumns رشته
اسم ستون‌ها که با استفاده از کاما از هم جدا شده اند. استفاده از آن اختیاری است و البته منسوخ هم شده است و در نسخه‌های جدید jQuery DataTables از آن پشتیبانی نمی‌شود.

aaData آرایه
همان طور که قبلا هم گفتیم، مقادیر سلول هایی را که باید در جدول نشان داده شوند را در خود نگهداری می‌کند. یعنی در واقع داده‌های جدول در آن ریخته می‌شوند. هر وقت که DataTables داده‌های مورد نیازش را دریافت می‌کند، سلول‌های جدول html مربوطه اش را از روی آرایه aaData ایجاد می‌کند. تعداد ستون‌ها در این آرایه دو بعدی، باید با تعداد ستون‌های جدول html مربوطه به آن یکسان باشد

شکل زیر پارامترها دریافتی از سرور را نشان می‌دهند:



شکل ب) پارامترهای دریافتی از سرور به صورت json

استفاده از روش server side processing در mvc

همان طور که گفتیم، کلاینت به سرور یک سری پارامترها را ارسال می‌کند و آن پارامترها را هم شرح دادیم. برای دریافت این پارامترها طرف سرور، احتیاج به یک مدل هست. این مدل به صورت زیر پیاده سازی خواهد شد:

```
/// <summary>
/// Class that encapsulates most common parameters sent by DataTables plugin
/// </summary>
public class jQueryDataTableParamModel
{
    /// <summary>
    /// Request sequence number sent by DataTable,
    /// same value must be returned in response
    /// </summary>
    public string sEcho { get; set; }
    /// <summary>
    /// Text used for filtering
    /// </summary>
    public string sSearch { get; set; }
    /// <summary>
    /// Number of records that should be shown in table
    /// </summary>
    public int iDisplayLength { get; set; }
    /// <summary>
    /// First record that should be shown(used for paging)
    /// </summary>
    public int iDisplayStart { get; set; }
    /// <summary>
    /// Number of columns in table
    /// </summary>
    public int iColumns { get; set; }
    /// <summary>
    /// Number of columns that are used in sorting
    /// /// </summary>
    public int iSortingCols { get; set; }
    /// <summary>
    /// Comma separated list of column names
    /// </summary>
    public string sColumns { get; set; }
}
```

مدل بایندر mvc وظیفه مقداردهی به خصوصیات درون این کلاس را بر عهده دارد، بقیه پارامترهایی که به سرور ارسال می‌شوند و در این کلاس نیامده اند، از طریق شیء Request در دسترس خواهند بود.

اکشن متدی که مدل بالا را دریافت می‌کند، می‌تواند به صورت زیر پیاده سازی شود. این اکشن متد وظیفه پاسخ دادن به درخواست DataTables بر اساس پارامترهای ارسال شده در مدل DataTablesParam را دارد. خروجی این اکشن متد شامل پارارمترهای مورد نیاز پلاگین DataTables برای تشکیل جدول است که آنها را هم شرح دادیم.

```
public JsonResult GetBrowsers(jQueryDataTableParamModel param)
{
        IQueryable<Browser> allBrowsers = new Browsers().CreateInMemoryDataSource().AsQueryable();

        IEnumerable<Browser> filteredBrowsers;

        // Apply Filtering
        if (!string.IsNullOrEmpty(param.sSearch))
        {
                filteredBrowsers = new Browsers().CreateInMemoryDataSource()
                    .Where(x => x.Engine.Contains(param.sSearch)
                                || x.Grade.Contains(param.sSearch)
                                || x.Name.Contains(param.sSearch)
                                || x.Platform.Contains(param.sSearch)
                    ).ToList();
                float f;
                if (float.TryParse(param.sSearch, out f))
                {
                    filteredBrowsers = filteredBrowsers.Where(x => x.Version.Equals(f));
                }
        }
        else
        {
                filteredBrowsers = allBrowsers;
        }

        // Apply Sorting
        var sortColumnIndex = Convert.ToInt32(Request["iSortCol_0"]);
        Func<Browser, string> orderingFunction = (x => sortColumnIndex == 0 ? x.Engine :
                                                       sortColumnIndex == 1 ? x.Name :
                                                       sortColumnIndex == 2 ? x.Platform :
                                                       sortColumnIndex == 3 ? x.Version.ToString()
:
                                                       sortColumnIndex == 4 ? x.Grade :
                                                           x.Name);

        var sortDirection = Request["sSortDir_0"]; // asc or desc
        filteredBrowsers = sortDirection == "asc" ? filteredBrowsers.OrderBy(orderingFunction) :
filteredBrowsers.OrderByDescending(orderingFunction);

        // Apply Paging
        var enumerable = filteredBrowsers.ToArray();
        IEnumerable<Browser> displayedBrowsers = enumerable.Skip(param.iDisplayStart).
                Take(param.iDisplayLength).ToList();

        return Json(new
        {
                sEcho = param.sEcho,
                iTotalRecords = allBrowsers.Count(),
                iTotalDisplayRecords = enumerable.Count(),
                aaData = displayedBrowsers
        }, JsonRequestBehavior.AllowGet);
}
```

تشریح اکشن متد GetBrowsers :

این اکشن متد از مدل jQueryDataTableParamModel به عنوان پارامتر ورودی خود استفاده می‌کند. این مدل همان طور هم که گفتیم، شامل یک سری خصوصیت است که توسط پلاگین jQuery DataTables مقداردهی می‌شوند و همچنین مدل بایندر mvc وظیفه بایند کردن این مقادیر به خصوصیات درون این کلاس را بر عهده خواهد داشت. درون بدنه اکشن متد GetBrowsers داده‌ها بعد از اعمال عملیات فیلترینگ، مرتب سازی، و صفحه بندی به فرمت مناسبی درآمده و به طرف مرورگر فرستاده خواهند شد.

برای پیاده سازی کدهای طرف کلاینت نیز، درون یک View کدهای زیر قرار خواهند گرفت:

```
$(function () {
        var $table = $('#browsers-grid');
        $table.dataTable({
                "bProcessing": true,
                "bStateSave": true,
                "bServerSide": true,
                "bFilter": true,
                "sDom": 'T<"clear">lftipr',
                "aLengthMenu": [[5, 10, 25, 50, -1], [5, 10, 25, 50, "All"]],
                "bAutoWidth": false,
                "sAjaxSource": "/Home/GetBrowsers",
                "fnServerData": function (sSource, aoData, fnCallback) {
                    $.ajax({
                        "dataType": 'json',
                        "type": "POST",
                        "url": sSource,
                        "data": aoData,
                        "success": fnCallback
                    });
                },
                "aoColumns": [
                    { "mDataProp": "Engine" },
                    { "mDataProp": "Name" },
                    { "mDataProp": "Platform" },
                    { "mDataProp": "Version" },
                    { "mDataProp": "Grade" }
                ],
                "oLanguage": {
                        "sUrl": "/Content/dataTables.persian.txt"
                }
        });
});
```

تشریح کدها:

fnServerData :

این متد، در واقع نحوه تعامل سرور و کلاینت را با استفاده از درخواستهای XHR مشخص خواهد کرد.

oLanguage :

برای فعال سازی زبان فارسی، فیلدهای مورد نیاز ترجمه شده و در یک فایل متنی قرار داده شده اند. کافی است آدرس این فایل متنی به ویژگی oLanguage اختصاص داده شوند.

مثال این قسمت را از لینک زیر دریافت کنید:
DataTablesTutorial04.zip

لازم به ذکر است پوشه obj، bin، و packages جهت کاهش حجم این مثال از solution حذف شده اند. برای اجرای این مثال از اینجا کمک بگیرید.

مطالعه بیشتر

برای مطالعه بیشتر در مورد این پلاگین و نیز پیاده سازی آن در MVC می‌توانید به لینک زیر نیز مراجعه بفرمائید که بعضی از قسمتهای این مطلب هم از مقاله زیر استفاده کرده است:
jQuery DataTables and ASP.NET MVC Integration - Part I

نظرات خوانندگان

نویسنده: Sorosh
تاریخ: ۱۳۹۲/۱۱/۱۱ ۸:۴۹

با سلام و تشکر. می‌خوام یک ستون ردیف به این جدول اضافه کنم که به ازای صفحه، ردیف جلو بره یعنی از 1 نشه دوباره و یکی هم اضافه کردن یک Attr خاص و جدید به هر سطر در جدول مثل ProjectCode که اون هم داخلش Id اون سطر در دیتا بیس هستش . ممنونم اگه کمکم کنید
کد کارم رو هم می‌دهم . البته من در WebForm کار کردم.

```
var $table = $('#browsers-grid');
$table.dataTable({
"bJQueryUI": true,
"bProcessing": true,
"bSortClasses": false,
"bServerSide": true,
"bFilter": true,
"sPaginationType": "full_numbers",
"sScrollY": 400,
"sScrollX": "100%",
"sScrollXInner": "110%",
"bLengthChange": false,
"iDisplayLength": 20,

"aLengthMenu": [[5, 10, 25, 50, -1], [5, 10, 25, 50, "All"]],
"bAutoWidth": false,
"sAjaxSource": "Commands.aspx?cmd=all",
"fnServerData": function (sSource, aoData, fnCallback) {
  $.ajax({
"dataType": 'json',
"type": "POST",
"url": sSource,
"data": aoData,
"success": fnCallback
  });
},

"aoColumns":
  [

  { "mDataProp": "Code" },
  { "mDataProp": "Caption" },
  { "mDataProp": "Comment" }
  ],

"oLanguage":
  {
"sUrl": "dataTables.persian.txt"
  }
});
```

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۱۱/۱۱ ۹:۴۷

سلام، خواهش می‌کنم. می‌تونید در سمت سرور بعد از واکشی اطلاعات از دیتابیس اونو داخل یک منبع داده درون حافظه ای بریزید و هر تعداد ستون که لازم دارید به اون منبع داده جدید اضافه کنید. و با مقدارهای مناسبی هر مدخل از اون منبع داده رو پر کنید. مثلا فرضا اگه جدول دیتابیس شما دارای سه ستون Code و Caption و Comment هست کلاس جدیدی بسازید که این سه تا ستون رو داره (به عنوان پروپرتی) و پروپرتی‌های دلخواه دیگه ای هم داره. مثلا پروپرتی RowNumber . بعد لیستی از داده‌ها رو که از دیتابیس واکشی کردید داخل لیستی از ViewModel ساخته شده بریزید و خصوصیت RowNumber هر ViewModel رو مقداردهی مناسبی کنید.

نویسنده: Sorosh
تاریخ: ۱۳۹۲/۱۱/۱۱ ۱۱:۶

متشکرم از پاسخ شما . البته اگه می‌شد نمونه‌ای ایجاد کنید خیلی بهتر بود که باز هم ممنونم. اما فقط برای بحث خصوصیت هر

سطر چی؟ می‌دونید اگه بخواهیم به هر سطر یک Attr جدید بدهیم چطوری باید اونو به TR نسبت بدهیم. مثلا من می‌خواهم به ازای هر سطر <tr/><'12'=tr ProjectCode> داشته باشم.

نویسنده: Sorosh
تاریخ: ۱۳۹۲/۱۱/۱۱ ۱۳:۱

با سلام مجدد؛ بالاخره پیداش کردم. برای ایجاد خصوصیت جدید به ازاء هر سطر و گرفتن مقدار از aoData بصورت زیر است .
مثلا برای اضافه کردن Attr - StudentCode

```
fnRowCallback": function (nRow, aData, iDisplayIndex)
{

  var StudentCode= aData["Code"];
  $(nRow).attr("StudentCode",StudentCode);
  return nRow;
}
```

با تشکر از شما

نویسنده: sorosh
تاریخ: ۱۳۹۲/۱۱/۱۲ ۱۳:۳۱

با سلام؛ من می‌خوام با قراردادن یک دکمه روی فرم به نام به‌روز رسانی، خاصیت اطلاعات داخل کوکی که با متد bStateSave کار می‌کنه از بین بره و جدول دوباره از نو رفرش بشه و حالت‌های جستجو و مرتب سازی قبلی از بین بره. متشکرم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۲ ۱۴:۲

- راه حل رسمی برای حذف کوکی ندارد . فقط مدت زمان آن قابل تنظیم است.
- به مطلب « کوکی در جاوا اسکریپت » در مورد حذف کوکی در همان سمت کلاینت مراجعه کنید. نام کوکی‌ها را هم با استفاده از افزونه Cookies manager می‌توانید مشاهده کنید.

نویسنده: رویا حیدری
تاریخ: ۱۳۹۳/۱۰/۱۶ ۱۶:۳۱

با سلام و تشکر فراوان. من میخواستم یک ستون با عنوان عملیات به جدول اضافه کنم که به صورت منو باشه و عملیات مختلف مثل ویرایش و حذف و تغییرات دیگه را از اون منو انتخاب و انجام بدم. برای اینکار باید چه کاری انجام بدم؟ و اینکه آیا برای اضافه کردن امکان ویرایش باید حتما از Editable DataTable استفاده کنم یا میشه تو همین نوع ساده هم امکان ویرایش را اضافه کرد؟ (Inline Edit نمی‌خوام انجام بدم)
ممنون میشم اگه راهنمایی کنید.

نویسنده: رویا حیدری
تاریخ: ۱۳۹۳/۱۰/۱۷ ۱۵:۰

در این صفحه تا حدودی به جوابم رسیدم.