

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.

اسکرپتی 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 پیام 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 مثلاً 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"
```

رشته : *sPaginationType*

نحوه صفحه بندی و حرکت بین صفحات مختلف را بیان می‌کند. اگر با `two_button` مقدار دهی شود (مقدار پیش فرض) حرکت بین صفحات مختلف به وسیله دکمه‌های `Next` و `Previous` امکان پذیر خواهد بود. اگر با `full_numbers` مقدار 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.



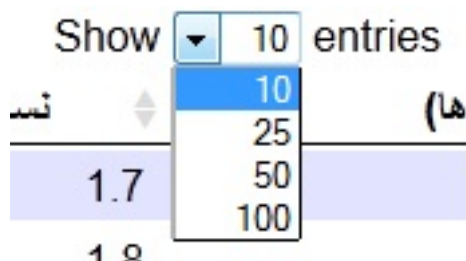
شکل الف) صفحه بندی به صورت `full_numbers`

بولین : *bLengthChange*

بیان می‌کند کاربر بتواند اندازه صفحه را تغییر دهد 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، مثلاً

```
"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 `value` و `text` این عناصر با هم فرق کند از یک آرایه دو بعدی استفاده خواهیم کرد، مثلاً:

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

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، مثلاً 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، مثلاً:

```
'<"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 x Showing 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، f، و l و p درون آن قسمت‌های اطلاعات، کادر جستجو، 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 سر حرف table برای جدول حاوی داده‌ها

i سر حرف information برای قسمت Showing x to xx from xxx entries

p سر حرف pagination برای قسمت صفحه بندی

r حرف دوم pProcessing برای قسمت پیغام قبل از بار کردن داده‌های جدول (قسمت loading)

H و F مربوط به themeهای jQuery UI می‌شوند که بعداً 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					
<div> <div>Show</div> <div> <div>1</div> <div>2</div> <div>3</div> <div>Next</div> <div>Last</div> </div> <div> <div>First</div> <div>Previous</div> <div>Next</div> <div>Last</div> </div> </div>					
نمبره CSS	نسخه موتور	پلتفرم (ها)	مرورگر	موتور رندرگیری	
A	1.7	+Win 98+ / OSX.2	Firefox 1.0	Gedko	
A	1.8	+Win 98+ / OSX.2	Firefox 1.5	Gedko	
A	1.8	+Win 98+ / OSX.2	Firefox 2.0	Gedko	
A	1.9	+Win 2k+ / OSX.3	Firefox 3.0	Gedko	
A	1.8	+OSX.2	Camino 1.0	Gedko	
A	1.8	+OSX.3	Camino 1.5	Gedko	
A	1.7	Win 95+ / Mac OS 8.6-9.2	Netscape 7.2	Gedko	
A	1.7	+Win 98SE	Netscape Browser 8	Gedko	
A	1.8	+Win 98+ / OSX.2	Netscape Navigator 9	Gedko	
A	1	+Win 95+ / OSX.1	Mozilla 1.0	Gedko	
A	1.1	+Win 95+ / OSX.1	Mozilla 1.1	Gedko	
A	1.2	+Win 95+ / OSX.1	Mozilla 1.2	Gedko	
A	1.3	+Win 95+ / OSX.1	Mozilla 1.3	Gedko	
A	1.4	+Win 95+ / OSX.1	Mozilla 1.4	Gedko	
A	1.5	+Win 95+ / OSX.1	Mozilla 1.5	Gedko	
A	1.6	+Win 95+ / OSX.1	Mozilla 1.6	Gedko	
A	1.7	+Win 98+ / OSX.1	Mozilla 1.7	Gedko	
A	1.8	+Win 98+ / OSX.1	Mozilla 1.8	Gedko	
A	1.8	+Win 98+ / OSX.2	Seamonkey 1.1	Gedko	
A	1.8	Gnome	Epiphany 2.20	Gedko	
C	3.1	KDE 3.1	Konqueror 3.1	KHTML	
A	3.3	KDE 3.3	Konqueror 3.3	KHTML	
A	3.5	KDE 3.5	Konqueror 3.5	KHTML	
C	-	Embedded devices	NetFront 3.1	Misc	
A	-	Embedded devices	NetFront 3.4	Misc	
نمبره CSS	نسخه موتور	پلتفرم (ها)	مرورگر	موتور رندرگیری	
Showing 1 to 25 of 57 entries					
<div> <div>Show</div> <div> <div>1</div> <div>2</div> <div>3</div> <div>Next</div> <div>Last</div> </div> <div> <div>First</div> <div>Previous</div> <div>Next</div> <div>Last</div> </div> </div>					

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: ۷:۴۷ ۱۳۹۲/۰۴/۰۷

با سلام و عرض ادب

زمانیکه من با Ajax , JQuery سطرهای دیتای جدول مورد نظر برای DataTable شدن را از سمت سرور ایجاد می‌کنم متأسفانه 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();
}
```

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 در 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.