

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 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.



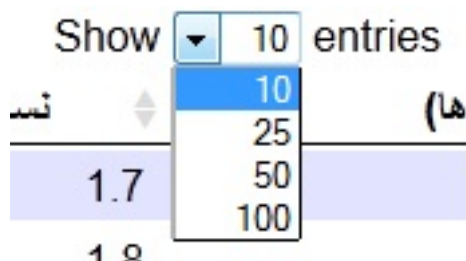
شکل الف) صفحه بندی به صورت `full_numbers`

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.

آرایه یک بعدی یا دو بعدی : *aLengthMenu*

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 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، 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> <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> <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](http://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.





✿ Add Note

می‌خواهیم با تغییر [jQuery Image Annotation](#) این پلاگین و برای asp.net استفاده کنیم

### ایجاد دیتابیس

ابتدا یک دیتابیس به نام Coordinates ایجاد کنید و سپس جدول زیر رو ایجاد کنید

```
USE [Coordinates]
GO
CREATE TABLE [dbo].[Coords2](
  [top] [int] NULL,
  [left] [int] NULL,
  [width] [int] NULL,
  [height] [int] NULL,
  [text] [nvarchar](50) NULL,
  [id] [uniqueidentifier] NULL,
  [editable] [bit] NULL
) ON [PRIMARY]
GO
```

## ایجاد کلاس Coords برای خواندن و ذخیره اطلاعات

```
public class Coords
{
    public string top;
    public string left;
    public string width;
    public string height;
    public string text;
    public string id;
    public string editable;

    public Coords(string top, string left, string width, string height, string text, string id, string
editable)
    {
        this.top = top;
        this.left = left;
        this.width = width;
        this.height = height;
        this.text = text;
        this.id = id;
        this.editable = editable;
    }
}
```

فرم اصلی برنامه شامل 3 وب سرویس به شرح زیر می باشد

## 1-GetDynamicContext

این متد در زمان لود اطلاعات از دیتابیس استفاده می شود (وقتی که postback صورت می گیرد)

```
[WebMethod]
public static List<Coords> GetDynamicContext(string entryId, string entryName)
{
    List<Coords> CoordsList = new List<Coords>();

    string connect = "Connection String";
    using (SqlConnection conn = new SqlConnection(connect))
    {
        string query = "SELECT [top], [left], width, height, text, id, editable FROM Coords2";
        using (SqlCommand cmd = new SqlCommand(query, conn))
        {
            conn.Open();
            using (SqlDataReader reader=cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    CoordsList.Add(new Coords(reader["top"].ToString(), reader["left"].ToString(),
reader["height"].ToString(),
                    reader["width"].ToString(),
                    reader["text"].ToString(), reader["id"].ToString(),
                    reader["editable"].ToString()));
                }
            }
            conn.Close();
        }
    }

    return CoordsList;
}
```

## DeleteCoords و SaveCoords- 2,3

این دو متد هم واسه ذخیره و حذف می باشند که نکته خاصی ندارند و خودتون بهینه اش کنید (در فایل ضمیمه موجودند)

## تغییر فایل jquery.annotate.js جهت فراخوانی وب سرویس ها

فقط لازمه که سه قسمت زیر رو در فایل اصلی تغییر بدید

```
$.fn.annotateImage.ajaxLoad = function (image) {
    ///<summary>
    ///Loads the annotations from the "getUrl" property passed in on the
    /// options object.
    ///</summary>

    $.ajax({
        type: "POST",
        contentType: "application/json; charset=utf-8",
        url: "Default.aspx/GetDynamicContext",
        data: "{ 'entryId': '" + 1 + "', 'entryName': '" + 2 + "'" },",
        dataType: "json",
        success: function (msg) {
            image.notes = msg.d;
            $.fn.annotateImage.load(image);
        }
    });
};
```

```
$.fn.SaveCoords = function (note) {
    $.ajax({
        type: "POST",
        contentType: "application/json; charset=utf-8",
        url: "Default.aspx/SaveCoords",
        data: "{ 'top': '" + note.top + "', 'left': '" + note.left + "', 'width': '" + note.width +
        "', 'height': '" + note.height + "', 'text': '" + note.text + "', 'id': '" + note.id + "', 'editable': '" +
        note.editable + "'" },",
        dataType: "json",
        success: function (msg) {
            note.id = msg.d;
        }
    });
};
```

```
$.fn.annotateView.prototype.edit = function () {
    ///<summary>
    ///Edits the annotation.
    ///</summary>
    if (this.image.mode == 'view') {
        this.image.mode = 'edit';
        var annotation = this;
        // Create/prepare the editable note elements
        var editable = new $.fn.annotateEdit(this.image, this.note);
        $.fn.annotateImage.createSaveButton(editable, this.image, annotation);
        // Add the delete button
        var del = $('<a>حذف</a>');
        del.click(function () {
            var form = $('#image-annotate-edit-form form');
            $.fn.annotateImage.appendPosition(form, editable)
            if (annotation.image.useAjax) {
```

```
$.ajax({
    type: "POST",
    contentType: "application/json; charset=utf-8",
    url: "Default.aspx/DeleteCoords",
    //                                url: annotation.image.deleteUrl,
    //                                data: form.serialize(),
    data: '{"id': '"' + editable.note.id + '"}',
    dataType: "json",
    success: function (msg) {
        //                                image.notes = msg.d;
        //                                $.fn.annotateImage.load(image);
    },
    error: function (e) { alert("An error occurred deleting that note.") }
});

}

annotation.image.mode = 'view';
editable.destroy();
annotation.destroy();
});

editable.form.append(del);
$.fn.annotateImage.createCancelButton(editable, this.image);
}

};
```

این پروژه شامل یه سری فایل css هم هست که می‌تونید کل پروژه رو از [اینجا](#) دانلود کنید

## نظرات خوانندگان

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۳/۱۷ ۰:۲۲

ضمن تشکر، فقط نکته استفاده از [JSON.stringify](#) در حین کار با jQuery Ajax رو بهتره اعمال کنید تا در دراز مدت و حالت‌های مختلف ورودی به مشکل برنخورید. به صورت خلاصه اطلاعات ارسالی رو جمع‌نزد و تبدیل به رشته نکنید. یک [شیء کامل](#) درست کنید و اجازه بدید JSON.stringify اون رو تبدیل کنه.

نویسنده: یزدان  
تاریخ: ۱۳۹۲/۰۳/۱۸ ۱۰:۳۵

اگر متدهای وب سرویس رو درون صفحاتی قرار بدیم که نیاز به لاگین کردن و احراز هویت جهت دسترسی به آنها باشد، آیا میتوان متدهای وب سرویس رو خارج از اون صفحات فراخوانی نمود؟  
چه تضمینی وجود دارد که در خارج از صفحات با سطح دسترسی (حداقل لاگین) وب سرویسها فراخوانی نشوند؟  
به نظر شما MS AJAX کارش به پایان رسیده؟  
تشکر

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۳/۱۸ ۱۰:۵۴

- اگر دقت کرده باشید در کدهای فوق این متدها استاتیک تعریف شدن، یعنی مراحل چرخه طول عمر یک صفحه به آنها اعمال نشده و اصلا جزئی از مباحث اعتبارسنجی صفحه جاری لحاظ نخواهند شد.  
- در وب فرمها استفاده از وب متدها یک روش برای کار با jQuery Ajax است. روش دوم استفاده از Generic handler ها و فایل‌های ashx است. در این موارد به علت استاتیک نبودن handlerهای تولیدی، می‌شود همه نوع اعتبارسنجی رو اعمال کرد اعم از روش Forms Authentication مثلا توسط context.Request.IsAuthenticated یا حتی روش منسوخ شده استفاده از سشن‌ها برای اعتبارسنجی با پیاده‌سازی IRequiresSessionState.  
- در مطلب فوق اصلا از MS Ajax استفاده نشده. اون هم جایگاه خودش رو در کاربردهای خاص خودش دارد.

در این مطلب با نحوه استفاده از پلاگین جستجوی سفارشی searchboxmvc.js آشنا خواهید شد.



قبلاً در [اینجا](#) با نحوه ایجاد پلاگین jQuery آشنا شدید. روشی دیگری نیز برای ایجاد این نوع پلاگین‌ها وجود دارد و آن استفاده از [widget factory](#) موجود در پلاگین jQuery UI می‌باشد.

برای استفاده از این پلاگین که کدهای کامل آن در فایل پیوست موجود است، ابتدا باید فایل‌های لازم را به پروژه خود اضافه کنیم:

```
<link rel="stylesheet" href="@Url.Content("~/Content/bootstrap-rtl.css")" type="text/css" />
<script type="text/javascript" src="@Url.Content("~/scripts/jquery-2.0.2.min.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/jquery-ui-1.10.3.min.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/bootstrap-rtl.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/searchboxmvc.js")"></script>
```

سپس در کنترلر خود یک Action بصورت زیر ایجاد کنید:

```
[HttpPost]
public virtual ActionResult LoadData(string fieldName, string value, string stringFilterMode =
"startWith")
{
    Thread.Sleep(2000);
    var models = MakePersons();
    if (fieldName == "Id")
    {
        models = models.Where(p => p.Id == int.Parse(value)).Take(1).ToList();
    }
    else if (fieldName == "FirstName")
    {
        models = models.Where(p => p.FirstName.StartsWith(value)).ToList();
    }

    return Json(new { Status = "OK", Records = models });
}
private List<Person> MakePersons()
{
    var lst = new List<Person>();
    lst.Add(new Person() { Id = 1, Code = "Uytffs-098", FirstName = "احمد رضا", LastName =
"عابدزاده" });
    lst.Add(new Person() { Id = 2, Code = "fTuuw-652", FirstName = "کریم", LastName = "باقری"
});
    lst.Add(new Person() { Id = 3, Code = "Lopapo-123", FirstName = "خداداد", LastName =
"عزیزی" });
    lst.Add(new Person() { Id = 4, Code = "Utppq-981", FirstName = "علی", LastName = "دایی" });
    lst.Add(new Person() { Id = 5, Code = "zttsn-471", FirstName = "علی", LastName = "کریمی"
});
}
```

```
lst.Add(new Person() { Id = 6, Code = "poiud-901", FirstName = "مهدی", LastName = "مهدوی" });
lst.Add(new Person() { Id = 7, Code = "wqrPoP-391", FirstName = "علیرضا", LastName = "منصوریان" });
return lst;
}
```

در ادامه در ویوی مورد نظر خود یک div ایجاد کنید. همین div خام با اعمال پلاگین بر روی آن ، بصورت یک پلاگین جستجو عمل خواهد کرد.

حال کدهای جاوا اسکریپت مورد نظر را برای اعمال پلاگین و تنظیمات مورد نیاز آن به div ایجاد شده می‌نویسیم:

```
<div id="div_SearchBoxContainer">
</div>
<script type="text/javascript">
$(function() {
    $('#div_SearchBoxContainer').searchboxmvc({
        loadUrl: '@Url.Action(actionName: "LoadData", controllerName: "Home")',
        defaultStringFilterMode: "startWith",
        loadDataOnLeave: true,
        displayClass: "",
        displayNoResultClass: "",
        display: function (element, record) {
            $(element).html(record.FirstName + " " + record.LastName);
        },
        listItemsDisplay: function (element, record, index) {
            return record.LastName + " " + record.FirstName + "(" + record.Code + ")";
        },
        fields: [
            {
                fieldName: "Id",
                fieldTitle: "شناسه",
                width: 100,
                defaultValueField: true
            },
            {
                fieldName: "FirstName",
                fieldTitle: "نام",
                width: 200,
                defaultDisplayField: true,
                filter: true,
                isStringType: true
            },
            {
                fieldName: "LastName",
                fieldTitle: "نام خانوادگی",
                filter: false,
                isStringType: true
            }
        ]
    });
</script>
}
```

### شرح پارامترهای افزونه searchboxmvc.js

**loadUrl**: آدرس اکشن متدی است که بصورت ajax ای فراخوانی شده و نتایج حاصل را بازگشت میدهد. نتایج حاصله باید با فرمت json بازگشت داده شوند. اگر نتایج موفقیت باشد بصورت `Json(new { Status = "OK", Records = models })` بازگشت داده شوند و اگر خطایی در این بین صورت گرفت مقدار Status نباید مقدار OK باشد. پارامترهای مورد نیاز این اکشن نیز باید به ترتیب با نام های `fieldName` و `value` باشند که `fieldName` نام فیلدی است که جستجو بر اساس آن صورت می‌گیرد و `value` همان مقدار وارد شده توسط کاربر است. **defaultStringFilterMode**: اگر فیلد مورد جستجو از نوع رشته ای باشد (یعنی `isStringType` آن برابر true باشد) آنگاه پارامتر سوم اکشن متد بطور خودکار مقداردهی خواهد شد. مقادیر این خاصیت میتواند `startWith` یا `contains` یا `equal` باشد. **loadDataOnLeave**: اگر برابر false باشد، هر بار که متن input تغییر کرد بلافاصله یک تقاضا برای یافتن مقادیر به سرور فرستاده

میشود و نیازی نیست که فوکوس از کنترل خارج شود.

**displayClass** : نام کلاس css است که به div 3 اعمال خواهد شد.

**displayNoResultClass** : در صورتیکه جستجو نتیجه ای نداشته باشد این کلاس به div 3 اعمال خواهد شد.

**display** : یک فانکشن که برای ایجاد خروجی html برای نمایش در div 3 بکار می‌رود.

**listItemsDisplay** : یک فانکشن که برای ایجاد خروجی html برای آیتم‌ها بکار می‌رود.

**fields** : یک آرایه از فیلدهای موردنیاز پلاگین .

خاصیت‌های فیلد نیز بصورت زیر است:

**fieldName** : نام فیلد

**fieldTitle** : عنوان فیلد

**defaultValueField** : فیلد پیش فرض که جستجو بر اساس آن صورت می‌گیرد. اگر تعیین نشود فیلد اول آرایه به عنوان فیلد پیش فرض انتخاب خواهد شد.

**defaultDisplayField** : فیلد پیش فرض که برای نمایش متن div 3 بکار می‌رود(البته اگر پارامتر display تعیین نشود)

**filter** : اگر برابر true باشد این فیلد در لیست فیلدهای جستجو خواهد آمد و کاربر می‌تواند بر اساس آن جستجو انجام دهد.

**isStringType** : اگر برابر true باشد ، پارامتر سوم اکشن متد بطور خودکار مقداردهی خواهد شد.

لازم به ذکر است که این پلاگین کامل نیست و فقط برای ارائه مثال اینجا آورده شده است. هر یک از دوستان می‌توانند محتوای پلاگین را به سلیقه خود تغییر داده و پلاگین را کاملتر کنند.

[sample\\_mvc.zip](#)



## نظرات خوانندگان

نویسنده: ح مراداف  
تاریخ: ۱۶:۵۶ ۱۳۹۲/۱۱/۲۹

سلام ممنون از مطلب مفیدتون. چند تا سوال از خدمتتون داشتم: 1- آیا دیگه نیاز به قراردادن input نخواهیم داشت ؟ 2- فرض کنیم از این سیستم سرچ در بخشی از یک فرم استفاده کردیم و پس از کلیک روی دکمه submit نیاز داریم که به آی دی گزینه ای که توی بخش سرچ انتخاب شده ، دست پیدا کنیم ... چجوری باید به آی دی گزینه ای که توی کنترل سرچ انتخاب شده دست پیدا کنم ؟ با تشکر

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، مجموعه‌ای 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 سی‌اس‌اس (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 داده‌ها، مرتب سازی ستون‌ها، 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. مثلاً در کادر جستجو نوشت :

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

که یعنی 1 و 5 همه عددهای بین 1 و 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 sSearch0 ، sSearch1 ، sSearch2 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 ستون‌هایی 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`

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

A	1.7	+Win 98+ / OSX.2	Firefox 1.0	Gecko
A	1.8	+Win 98+ / OSX.2	Firefox 1.5	Gecko
A	1.8	+Win 98+ / OSX.2	Firefox 2	Gecko
A	1.8	+Win 98+ / OSX.2	Netscape Navigator 9	Gecko

```

{
  "sEcho": "2",
  "iTotalRecords": 17,
  "iTotalDisplayRecords": 17,
  "aaData": [
    {
      "Id": 15,
      "Engine": "Gecko",
      "Name": "Netscape Navigator 9",
      "Platform": "Win 98+ / OSX.2+",
      "Version": "1.8",
      "Grade": "A"
    },
    {
      "Id": 14,
      "Engine": "Gecko",
      "Name": "Netscape Browser 8",
      "Platform": "Win 98SE+",
      "Version": "1.7",
      "Grade": "A"
    },
    {
      "Id": 13,
      "Engine": "Gecko",
      "Name": "Netscape 7.2",
      "Platform": "Win 95+ / Mac OS 8.6-9.2",
      "Version": "1.7",
      "Grade": "A"
    },
    {
      "Id": 17,
      "Engine": "Gecko",
      "Name": "Mozilla 1.1",
      "Platform": "Win 95+ / OSX.1+",
      "Version": "1.1",
      "Grade": "A"
    },
    {
      "Id": 16,
      "Engine": "Gecko",
      "Name": "Mozilla 1.0",
      "Platform": "Win 95+ / OSX.1+",
      "Version": "1",
      "Grade": "A"
    },
    {
      "Id": 5,
      "Engine": "Trident",
      "Name": "Internet Explorer 7",
      "Platform": "Win XP SP2+",
      "Version": "7",
      "Grade": "A"
    },
    {
      "Id": 4,
      "Engine": "Trident",
      "Name": "Internet Explorer 6",
      "Platform": "Win98+",
      "Version": "6",
      "Grade": "A"
    },
    {
      "Id": 3,
      "Engine": "Trident",
      "Name": "Internet Explorer 5.5",
      "Platform": "Win95+",
      "Version": "5.5",
      "Grade": "A"
    },
    {
      "Id": 2,
      "Engine": "Trident",
      "Name": "Internet Explorer 5.0",
      "Platform": "Win95+",
      "Version": "5",
      "Grade": "C"
    },
    {
      "Id": 1,
      "Engine": "Trident",
      "Name": "Internet Explorer 4.0",
      "Platform": "Win95+",
      "Version": "4",
      "Grade": "X"
    },
    {
      "Id": 10,
      "Engine": "Gecko",
      "Name": "Firefox 3",
      "Platform": "Win 2k+ / OSX.3+",
      "Version": "1.9",
      "Grade": "A"
    },
    {
      "Id": 9,
      "Engine": "Gecko",
      "Name": "Firefox 2",
      "Platform": "Win 98+ / OSX.2+",
      "Version": "1.8",
      "Grade": "A"
    },
    {
      "Id": 8,
      "Engine": "Gecko",
      "Name": "Firefox 1.5",
      "Platform": "Win 98+ / OSX.2+",
      "Version": "1.5",
      "Grade": "A"
    },
    {
      "Id": 7,
      "Engine": "Gecko",
      "Name": "Firefox 1.0",
      "Platform": "Win 98+ / OSX.2+",
      "Version": "1.0",
      "Grade": "A"
    },
    {
      "Id": 6,
      "Engine": "Gecko",
      "Name": "Netscape Navigator 4",
      "Platform": "Win 95+ / OSX.2+",
      "Version": "4",
      "Grade": "A"
    },
    {
      "Id": 11,
      "Engine": "Gecko",
      "Name": "Netscape Navigator 3",
      "Platform": "Win 95+ / OSX.2+",
      "Version": "3",
      "Grade": "A"
    },
    {
      "Id": 12,
      "Engine": "Gecko",
      "Name": "Netscape Navigator 2",
      "Platform": "Win 95+ / OSX.2+",
      "Version": "2",
      "Grade": "A"
    }
  ]
}

```

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 پارامترهای 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 باینده 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 bin، obj 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: ۸:۴۹ ۱۳۹۲/۱۱/۱۱

با سلام و تشکر. می‌خواهم 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 کمک 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: پیمان پارسائی

تاریخ: ۹:۴۷ ۱۳۹۲/۱۱/۱۱

سلام، خواهش می‌کنم. می‌تونید در سمت سرور بعد از واکنشی 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. مثلاً فرضاً 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. مثلاً پروپرتی RowNumber. بعد لیستی از داده‌ها رو که از دیتابیس واکنشی کردید داخل لیستی از ViewModel ساخته شده بریزید و خصوصیت RowNumber هر ViewModel رو مقداردهی مناسبی کنید.

نویسنده: Sorosh

تاریخ: ۱۱:۶ ۱۳۹۲/۱۱/۱۱

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 ProjectCode='12'></tr>` داشته باشم.

نویسنده: Sorosh

تاریخ: ۱۳:۱ ۱۳۹۲/۱۱/۱۱

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: ۱۳:۳۱ ۱۳۹۲/۱۱/۱۲

با سلام؛ من می‌خواهم 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: ۱۴:۲ ۱۳۹۲/۱۱/۱۲

- 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 [Cookies manager](#) می‌توانید مشاهده کنید.