

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 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 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
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 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


```

GET http://localhost:2080/Content/dataTables.persian.txt 304 Not Modified 37ms
POST http://localhost:2080/Home/GetBrowsers 200 OK 7ms
POST http://localhost:2080/Home/GetBrowsers 200 OK 19ms

Headers Post Response JSON Cookies
{
  "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 4.0",
      "Platform": "Win 95+ / OSX.2+",
      "Version": 4.0,
      "Grade": "A"
    },
    {
      "Id": 11,
      "Engine": "Gecko",
      "Name": "Netscape 3.0",
      "Platform": "Win 95+ / OSX.2+",
      "Version": 3.0,
      "Grade": "A"
    },
    {
      "Id": 12,
      "Engine": "Gecko",
      "Name": "Netscape 2.0",
      "Platform": "Win 95+ / OSX.2+",
      "Version": 2.0,
      "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.