عنوان: **بررسی چند کتابخانه آپلود با پشتیبانی از DragDrop**
نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۴/۳۰ ۱۳:۲۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, JavaScript, jQuery, FileUpload

برای یکی از پروژه‌ها نیاز به یک آپلودر داشتم که قابلیت Drag&Drop را نیز داشته باشد و در ضمن پیاده سازی آسانی هم داشته باشد. در این بین به تعدادی از کتابخانه‌های جی کوئری می‌پردازیم.

FileDrop

اولین کتابخانه‌ای که با آن آشنا شدم و از آن استفاده کردم، کتابخانه‌ی FileDrop است که بسیار ساده و در عین حال قابلیت‌های خوبی را می‌دهد و از فناوری Filereader (+) در Html5 برای اینکار استفاده می‌کند. مرورگرهای کروم، فایرفاکس 3.6 به بعد، IE10 به بعد و Opera 12 به بعد از آن پشتیبانی می‌کنند.

فایل‌های مورد نیاز را از اینجا دانلود کنید . فایل اسکریت آن را ابتدا صدا بزنید:

```
<script src="~/scripts/jquery.filedrop.js" type="text/javascript"></script>
```

سپس المان‌های زیر را به کد HTML خود اضافه کنید:

```
<div id="dropZone">فایل برنامه را به داخل این کادر بکشانید</div>
        <br>
        فایل یا فایل‌های آپلود شده:
        <ul id="uploadResult"></ul>
```

تگ اول، محلی است که فایل‌ها به سمت آن درگ و روی آن دراپ می‌شوند که از این به بعد به آن محل آپلود می‌گوییم. المان بعدی جهت گزارش فایل‌هایی است که آپلود شده‌اند. با آپلود شدن هر تعداد فایل، اسم آن به لیست اضافه می‌گردد.
کدهای css زیر را هم به صفحه اضافه کنید تا محل آپلود زیباتر شود:

```
.files {
            min-height: 42px;
            background: #CCC none repeat scroll 0% 0%;
            border-top: 1px solid #FFF;
            margin: 11px 0px;
            padding: 11px 13px;
            border-radius: 6px;
        }

        #dropZone.mouse-over {
            background-color: #1d4257;
        }
```

کد جی کوئری زیر را به صفحه اضافه کنید:

```
$('#dropZone').filedrop({
                url: uploadAddress,
                paramname: 'files',
                maxFiles: 1,
                dragOver: function() {
                    $('#dropZone').addClass('mouse-over');
                },
                dragLeave: function() {
                    $('#dropZone').removeClass('mouse-over');
                },
                drop: function() {
                    $('#dropZone').removeClass('mouse-over');
                },
                afterAll: function() {
                    $('#dropZone').html('آپلود با موفقیت انجام شد');
```

```
            },
            uploadFinished: function(i, file, response, time) {
                $('#uploadResult').append('<li>' + file.name + '</li>');
            }
        });
```

ابتدا پلاگین جی کوئری را روی تگ مربوطه اعمال می‌کنیم و سپس پارامترها را با مشخصات زیر اعمال می‌کنیم:

Url	آدرسی که قرار است فایل‌ها به آن سمت ارسال شوند.
Paramname	در سمت سرور باید فایل‌ها را با استفاده از این نام پارامتر دریافت کنید.
maxFiles	تعداد فایلهایی که میتوان با درگ و دراپ کردن روی آن به دست آورد. در بالا به یک فایل محدود شده است.
dragOver	این رویداد زمانی اجرا خواهد شد که اشاره گر با حالت درگ کرده فایل‌ها را به محل آپلود آورده است.
dragLeave	موقعی که ماوس از محل آپلود خارج می‌شود
drop	موقعی که شما فایل‌ها را روی محل آپلود رها می‌کنید.
afterAll	بعد از اینکه همه کارها تمام شد اجرا می‌شود.(آخرین رویداد)
uploadFinished	کار آپلود به پایان رسیده است. در مثال بالا پس از پایان آپلود، نام فایل آپلود شده را به کاربر نشان داده‌ایم.

نحوه‌ی دریافت آن در سمت سرور, در یک اکشن متد به صورت زیر است:

```
[HttpPost]
        public virtual ActionResult UpdateApp(IEnumerable<HttpPostedFileBase>files)
        {
            foreach (HttpPostedFileBase file in files)
            {
                string filePath = Path.Combine(TempPath, file.FileName);
                file.SaveAs(filePath);
            }

            return Json(new {state = "success", message = "با موفقیت عملیات ارسال فایل انجام شد"},
JsonRequestBehavior.AllowGet);
        }
```

در اکشن متد بالا ما فایل‌ها را از طریق نام پارامتر `files` که مشخص کرده بودیم، به عنوان یک لیست شمارشی دریافت می‌کنیم. کدها بالا برای ساده‌ترین راه اندازی ممکن کفایت می‌کنند.

این موارد از اصلی‌ترین‌ها هستند که به کار می‌آیند. به غیر این‌ها یک سری خصوصیات اضافه‌تری هم برای آن وجود دارد.

fallback_id	اگر دوست دارید این آپلودر را نیر به یک آپلودر معمولی اتصال دهید از این شناسه استفاده کنید.
withCredentials	با استفاده از کوکی‌ها یک درخواست cross-origin ایجاد می‌کند.
data	اگر دوست دارید به همراه فایل‌ها اطلاعات دیگری هم به همراه آن ارسال و پست شوند از این طریق اقدام نمایید. می‌تواند در قالب یک متغیر باشد یا خروجی یک تابع.

fallback_id	اگر دوست دارید این آپلودر را نیر به یک آپلودر معمولی اتصال دهید از این شناسه استفاده کنید.
	`` data: { param1: 'value1', param2: function(){ return calculated_data; } ```
headers	برای ارسال مقدار اضافه‌تر در هدر درخواست به کار میرود و صدا زدن آن همانند کد data می‌باشد.
error	در صورتیکه در فرایند آپلود خطایی رخ دهد، اجرا می‌گردد. نحوه‌ی کدنویسی آن و بررسی خطاهای آن به شرح زیر است: ``` error: function(err, file) { switch(err) { case 'BrowserNotSupported': alert('مرورگر از این فناوری پشتیبانی نمی‌کند') break; case 'TooManyFiles': // قصد آپلود همزمان فایل‌های بیشتری از حد مجاز تعیین شده دارید break; case 'FileTooLarge': // حداقل حجم یکی از فایل‌ها از حجم مجاز تعیین شده بیشتر است // برای دسترسی به نام آن فایل از کد زیر استفاده کنید //file.name break; case 'FileTypeNotAllowed': // نوع حداقل یکی از فایل‌ها با نوع‌ها مشخص شده ما یکی نیست break; case 'FileExtensionNotAllowed': // پسوند حداقل یکی از فایل‌ها مورد تایید نیست break; default: break; } } ```
allowedfiletypes	نوع فایل‌های مجاز را تعیین می‌کند: ``` allowedfiletypes: ['image/jpeg','image/png','image/gif'] ```
allowedfileextensions	پسوند فایل هایی که برای آپلود مجاز هستند را معرفی می‌کند. ``` allowedfileextensions: ['.apk','.jar'] ```
maxfilesize	حداکثر حجم مجاز برای هر فایل که به مگابایت بیان می‌شود.
docOver	این رویداد زمانی اجرا می‌شود که فایل‌های درگ شده شما وارد محیط یا پنجره مرورگر می‌شود.
uploadStarted	این رویداد زمانی اجرا میگردد که فرایند آپلود هر فایل به طور جداگانه در حال آغاز شدن است: متغیر i در کد زیر شامل اندیس فایلی است که آپلودش آغاز شده است و این اندیس از صفر آغاز می‌شود. متغیر file دسترسی شما را به اطلاعات یک فایل باز میکند

fallback_id	اگر دوست دارید این آپلودر را نیر به یک آپلودر معمولی اتصال دهید از این شناسه استفاده کنید.
	مانند نام فایل. متغیر len تعداد فایل هایی را که کاربر در محل آپلود رها کرده است، باز میگرداند. `function(i, file, len){ },`
uploadFinished	با اتمام آپلود هر فایل، این رویداد فراخوانی می‌گردد. دو پارامتر اول آن، همانند سابق هستند. پارامتر response خروجی json ایی را که در سمت سرور برگرداندیم، به ما باز می‌گرداند. پارامتر بعدی، زمانی را که برای آپلود طول کشیده است، بر می‌گرداند. `function(i, file, response, time) { }`
progressUpdated	این رویداد برای نمایش پیشرفت یک آپلود مناسب است که آخرین پارامتر آن یک عدد صحیح از پیشرفت فایل را بر می‌گرداند. `function(i, file, progress) { },`
globalProgressUpdated	این رویداد میزان پیشرفت کلیه فایل‌ها را به درصد باز می‌گرداند: `function(progress) { $('#progress div') .width(progress+"%"); }`
speedUpdated	سرعت آپلود هر فایل را با کیلوبیت بر ثانیه مشخص می‌کند. `function(i, file, speed) { }`
rename	در صورتی که قصد تغییر نام فایل ارسالی را دارید می‌توانید از این رویداد استفاده کنید. پارامتر name، نام اصلی فایل را بر می‌گرداند که می‌توانید آن را دستکاری کنید و نام جدیدی را به عنوان خروجی برگردانید. نمونه کاربردی از این رویداد `rename: function(name) { }`
beforeEach	این رویداد قبل از آپلود هر فایل آغاز می‌گردد و برگرداندن مقدار false در آن باعث جلوگیری و کنسل شدن آپلود آن فایل می‌گردد. `function(file) { }`

fallback_id	اگر دوست دارید این آپلودر را نیر به یک آپلودر معمولی اتصال دهید از این شناسه استفاده کنید.
beforeSend	پارامترهای اولی تکراری هستند ولی آخرین پارامتر یک تابع done را می‌توان به آن پاس کرد که قبل از اجرای کل عملیات آپلود صدا زده می‌شود. `function(file, i, done) {` `}`

رویدادی به اسم queuefiles هم هست تعداد فایل‌هایی را که میتوانند به طور موازی و همزمان آپلود گردند، مشخص می‌کند. ولی دراین حالت maxfiles مورد استفاده قرار نمی‌گیرد. جهت بررسی یک مثال عملی و همچنین کدهای سمت سرور در PHP میتوانید از این آموزش استفاده کنید. با تستی که به صورت لوکال رو آن انجام دادم به نظر نمی‌رسد برای فایل‌های با حجم متوسط به بالا مناسب باشد و برای فایل‌های با حجم کم مناسب می‌باشد. یک فایل 8 مگابایتی در حالت لوکال 9 ثانیه آپلود آن زمان برد و برای فایل‌های بزرگتر، فایرفاکس دیالوگ Stop Script را نشان داد.

PlUpload

این کتابخانه متن باز هم بسیار کارآمد و ساده و قابل انعطاف است و مثالهای آماده زیادی دارد. سایت سابسن هم در بخش آپلود زیرنویس‌ها از این کتابخانه استفاده می‌کند. از آنجا که آموزش این کتابخانه در سایت جاری آمده است از ذکر نکات بیشتر در مورد آن خودداری می‌نماییم.

Bootstrap FileStyle

اگر از قالب بوت استراپ استفاده می‌کنید و دوست دارید روی المان input file قدیمی، ولی به شکلی مدرن کار کنید این کتابخانه هم فراموش نشود.

DropZoneJS

این کتابخانه به نسبت DropFile امکانات بیشتری را دارد و در سایت اختصاصی آن مثال‌ها و مستندات خوبی قرار گرفته است. در ساده‌ترین حالت آن ابتدا فایل کتابخانه را صدا زده و سپس تگ فرم را به آن نسبت دهید:

```
<script src="https://rawgit.com/enyo/dropzone/master/dist/dropzone.js"></script>
<form action="/upload-target" class="dropzone"></form>
```

ولی اگر بخواهید آن را به سمت سرور ارسال کنید و از آنجا آن را کنترل کنید، کد فرم را به شکل زیر تغییر دهید:

ابتدا بسته‌ی نیوگت آن را صدا بزنید:

```
Install-Package dropzone
```

با نصب این کتابخانه یک سری فایل CSS هم به سیستم اضافه می‌شود که می‌توانید برای استایل دهی هر چه بیشتر از آن بهره ببرید. کد فرم را به شکل زیر تغییر دهید:

```
<form action="~/Home/SaveUploadedFile" method="post" enctype="multipart/form-data" class="dropzone"
id="dropzoneForm" style="width: 50px; background: none; border: none;">
        <div class="fallback">
            <input name="file" type="file" multiple />
            <input type="submit" value="Upload" />
        </div>
    </form>
```

تگی که با کلاس fallback مزین شده است موقعی به کار می‌آید که مرورگر از این کتابخانه پشتیبانی نکرده و مجبور به نمایش یک آپلود معمولی می‌شویم.

با استفاده از کدنویسی هم می‌توان یک المان را به یک آپلودر تبدیل کرد:

```
var myDropzone = new Dropzone("div#myId", { url: "/file/post"});

//============ OR ====================
$("div#myId").dropzone({ url: "/file/post" });
```

همانطور که می‌بینید الزامی برای اینکه از یک تگ فرم استفاده کنید ندارد.
برای کانفیگ آپلودرهایی که از طریف المانهای Html ایجاد می‌شوند، می‌توان از کد زیر استفاده کرد و یک تنظیم عمومی برای تمامی آپلودرهای html آن صفحه ایجاد کرد.

```
Dropzone.options.myId= {
  paramName: "file", //نام پارامتری که فایل از طریق آن انتقال می‌یابد
  maxFilesize: 2, // MB
  accept: function(file, done) {
    if (file.name == "justinbieber.jpg") {
      done("Naha, you don't.");
    }
    else { done(); }
  }
};
```

تابع بالا یک آرگومان از نوع file را برگردانده و اگر این تابع، تابع done را با پارامتری رشته‌ای صدا بزند، عملیات آپلود آن فایل کنسل شده و پیام خطایی را نمایش می‌دهد و در صورتیکه بدون پارامتر صدا زده شود، عمل آپلود بدون مشکل انجام می‌شود.
ازآنجا که این کتابخانه از تنظیمات وسیعی استفاده می‌کند و از حوصله‌ی این مقاله خارج است، بهتر هست که صفحه‌ی مستندات آن را که کامل هم هست، مطالعه بفرمایید. از سری قابلیت‌هایی که پشتیبانی می‌کند: موارد پوشش داده شده در FileDrop، ساخت layout، ایجاد صف، متد حذف و اضافه و از این قبیل، ایجاد تصویر تمبر مانند و ...
یک نکته تکمیلی در مورد *آپلود:* در ASP.net به طور پیش فرض نهایت حجم فایل آپلودی 4 مگابایتی تعیین شده است که میتوانید آن را از طریق web.config تغییر دهید:

```
<configuration>
    <system.web>
        <httpRuntime maxRequestLength="1048576" />
    </system.web>
</configuration>
```

برای IIS 7 به بعد هم از تکه کد زیر استفاده کنید:

```
<system.webServer>
   <security>
      <requestFiltering>
         <requestLimits maxAllowedContentLength="1073741824" />
      </requestFiltering>
   </security>
 </system.webServer>
```

در هر دو کد بالا نهایت حجم بر روی یک گیگابایت تعیین شده است که *maxRequestLength* به صورت کیلوبایت و *maxAllowContentLength* به صورت بایت تعیین شده است. توصیه می‌شود هر دو شکل آن را وارد کنید. به خصوص که IIS Express از کد ابتدایی استفاده می‌کند و بخواهید نتیجه‌ی آن را در تست‌ها ببینید.

نظرات خوانندگان

نویسنده: غلامرضا ربال
تاریخ: ۱۳۹۴/۰۴/۳۰ ۱۶:۴۱

این کتابخانه هم برای drag&drop جالب است. قابلیت ارسال اجکسی به صورت توکار ساپورت میشود.