

در یک پروژه من احتیاج داشتم تا عکس هایی که کاربر از طریق ckeditor آپلود می‌کند ، به صورت خودکار با lightbox یکپارچه شه و به صورت گالری عکس نمایش داده شود. همان طور که می‌دونید مکانیزم عملکرد اکثر پلاگین‌های lightbox به صورت زیر است:

```
<a href="original size of image directory"></a>
```

خب معلوم میشه که آدرس عکسی که به صورت کوچک (اصطلاحاً بند انگشتی) به نمایش در میاد باید داخل تگ img ، و آدرس عکس با سایز اصلی ، داخل تگ a قرار بگیره.

پس تقریباً معلوم شد که چه باید بکنیم.

ابتدا باید عکسی که آپلود شده را به صورت خودکار به اندازه کوچکتر تغییر ابعاد داده و سپس آدرس عکس تغییر ابعاد داده شده ، را به عنوان آدرس عکس آپلود شده به ckeditor باز گردانیم که در شکل زیر آن را نشان داده ام:



ولی ما فقط آدرس عکس تغییر ابعاد داده شده متعلق به تگ img را ، به ckeditor داده ایم و برای اینکه با lightbox به خوبی کار کند، احتیاج داریم که آدرس عکس با سایز اصلی را داخل تگ a قرار دهیم، که در ckeditor می‌توانیم از قسمت پیوندها این کار را انجام دهیم، و همه‌ی این عملیات باید به صورت خودکار انجام شود.



خب تا اینجا خلاصه ای از آنچه باید انجام شود را گفتیم.

در کل منطقی که باید پیاده شود بدین گونه است:

- 1- پیاده سازی ckeditor
- 2- فعال کردن قسمت آپلود عکس ckeditor
- 3- آپلود کردن عکس و تغییر اندازه آن به کمک کلاس WebImage
- 4- برگرداندن آدرس عکس‌های آپلود شده به ckeditor و روش‌های پیاده سازی آن

نکات:

الف) من در اینجا از ASP.NET MVC استفاده می‌کنم. شما نیز می‌توانید منطق پیاده سازی شده را به راحتی و با کمی تغییرات به پروژه خود اعمال کنید.

ب) کدهایی که من نوشتم 100% بهینه نیستند و کاملاً هم بدون اشکال نیستند ولی نیازهای شما را برآورده می‌کند.

ج) آدرس دادن فایل‌های من کاملاً صحیح نیستند و بهتر است شما از T4MVC استفاده کنید.

مرحله اول: دانلود و پیاده سازی ckeditor در صفحه

ابتدا به سایت ckeditor.com مراجعه و از قسمت دانلود، آن را دانلود کرده و سپس از حالت فشرده درآورده، در فولدری مثلاً به نام Scripts در پروژه‌ی خود بریزید.

همان طور که می‌دانید ckeditor با یک textarea یکپارچه می‌شود، که من ساده‌ترین آن را برای شما شرح می‌دهم. اول از همه شما باید یک فایل جاوا اسکریپت متعلق به ckeditor را در صفحه‌ای که از آن می‌خواهید استفاده کنید به صفحه ضمیمه کنید. نام این فایل ckeditor.js هست که در فایل داندودی در داخل پوشه‌ی ckeditor دیده می‌شود.

```
<script src="/Scripts/ckeditor/ckeditor.js" type="text/javascript"></script>
```

نکته: بهتر است که این ارجاع را قبل از بسته شدن تگ body انجام دهید یعنی درست قبل از </body>. خب فرض کنید که یک textarea به شکل زیر داریم. برای اینکه این ckeditor با textarea یکپارچه شود کافست که class آن را بر روی ckeditor قرار دهید.

```
<textarea id="content" name="content" class="ckeditor" ></textarea>
```

نکته: حتماً برای textarea خود id را نیز تعیین کنید، چون در غیر این صورت ckeditor کار نخواهد کرد.

تا به اینجا کار توانستیم ckeditor را اجرا کنیم. در مرحله بعد سراغ فعال سازی آپلود عکس در آن می‌رویم.

مرحله دوم: فعال سازی آپلود عکس در ckeditor

از شواهد این طور به نظر می‌آید که به صورت پیشفرض این امکان غیر فعال است و باید به صورت دستی آن را فعال کنیم. برای این کار کافست که از مستندات ckeditor کمک بگیریم. خب یکی از راحت‌ترین این روش‌ها این است که قبل از بسته شدن تگ بادی فایل جاوا اسکریپت زیر را بنویسیم:

```
<script type="text/javascript">
    CKEDITOR.replace( 'content' , { filebrowserImageUploadUrl: '/Admin/UploadImage' } );
</script>
```

نکته اول: این خط کدی که نوشتیم مرحله اول را باطل می‌کند یعنی احتیاج نبود که مرحله اول را طی کنیم و می‌توانستیم مستقیماً از همین مرحله شروع کنیم.

توضیحات:

آرگومان اول CKEDITOR.replace که در اینجا content است، در واقع id همان textarea ای هست که می‌خواهیم ckeditor روی آن اعمال شود.

آرگومان دوم نام کنترلر و اکشن را برای آپلود فایل مشخص می‌کند. (از منطق ASP.Net MVC استفاده کردم). خب تا اینجا اگر تست بگیرید می‌بینید که قسمت زیر برایتان فعال شده.



مرحله سوم: آپلود کردن عکس و تغییر اندازه آن به کمک کلاس WebImage

برای این که بینم که این فرم آپلود ckeditor ، چه پارامترهایی را به اکشن متد من ارسال می‌کند ، از فایرباگ کمک گرفتم:

```
<form lang="fa" action="/Admin
/UploadImage?CKEditor=Content&CKEditorFuncNum=2&
langCode=fa" dir="rtl" method="POST" enctype="multipart
/form-data">


```

همان طور که می‌بینید به خوبی آدرس post شدن اطلاعات به اکشن متدی که من برایش مشخص کردم را فهمیده. اما سه پارامتر دیگر نیز به اکشن متد ما ارسال می‌کند: CKEditor و CKEditorFuncNum و langCod. برای اینکه با این پارامترهای ارسالی بیشتر آشنا شوید ، توصیه می‌کنم [این صفحه](#) را ببینید. آنچه که از این پارامترها برای ما مهم هست ، من در اکشن متد تعریفی خود لحاظ کرده‌ام.

خب همون طور که پیش از این گفته بودم ما نیاز داریم که عکس آپلود شده را به ابعاد کوچکتر تغییر اندازه داده و اصطلاحاً از آن به عنوان تصویر بند انگشتی استفاده کنیم. آدرس این عکس کوچک شده ، همانیست که در آدرس تگ img قرار می‌گیرد و در ابتدا به کاربر نمایش داده می‌شود.

برای انجام این عمل من کلاسی را برای کار کردن با عکس برایتان معرفی می‌کنم، به نام WebImage که در داخل فضای نامی System.Web.Helpers قرار گرفته است.

از طریق این کلاس می‌توان کلیه عملیات دریافت فایل آپلود شده، ویرایش ، تغییر اندازه ، چرخاندن ، بریدن و حتی watermark کردن و در نهایت ذخیره عکس را به آسانی انجام داد.

من کدهای متد UploadImage را برایتان قرار می‌دهم که زیاد هم بهینه نیست و سپس برایتان توضیح می‌دهم.

```
public ActionResult UploadImage(string CKEditorFuncNum, string CKEditor, string langCode)
{
    string message; // message to display when file upload successfully(optional)
    string thumbPath = ""; // the directory for thumb file that should resize
    var db = new MyDbContext(); // make new instance from my context
    // here logic to upload image
    // and get file path of the image
    var file = WebImage.GetImageFromRequest(); // get the uploaded file from request
    var ext = Path.GetExtension(file.FileName); // get the path of file
    //get the file name without extension
```

```

var fileName = Path.GetFileNameWithoutExtension(file.FileName);
//add time to file name to avoid same name file overwrite, and then add extension to it
fileName += DateTime.Now.ToFileTime() + ext;
//choose the path for the original size of image
var path = Path.Combine(Server.MapPath("~/Content/UploadedImages/Default"), fileName);
file.Save(path); //save the original size of the image
db.Images.Add(new Image { RealName = file.FileName, FileName = fileName, UploadDate =
DateTime.Now }); // save image info to db
db.SaveChanges(); // submit changes to db
string defaultPath = "/Content/UploadedImages/Default/" + fileName; //path for original
size of //images
if (file.Width > 400) // if width of image bigger than 400 px do resize
{
    file.Resize(400, 400, true); //resize the image , third argument is aspect ratio
    string thumbName = "Thumb-" + fileName; // resized image name
    //path for resized image file
    string path2 = Server.MapPath("~/Content/UploadedImages/Thumbs/" + thumbName);
    thumbPath = "/Content/UploadedImages/Thumbs/" + thumbName;
    //save resized image file
    file.Save(path2);
}
else
{
    thumbPath = defaultPath; // if the size not bigger than 400px the thumb, path =
default path
}
// passing message success/failure
message = "Image was saved correctly";
// since it is an ajax request it requires this string
//java script that return files path to ckeditor
string output = @"<script>window.parent.CKEDITOR.tools.callFunction(" + CKEditorFuncNum +
", \" + thumbPath + "\", \" + message +
"\");window.parent.document.getElementById('cke_145_textInput').value=\"" + defaultPath +
";window.parent.document.getElementById('cke_125_textInput').value=0;</script>";
return Content(output);
}

```

توضیحات:

ابتدا به رشته به نام message در نظر گرفتیم ، برای هنگامی که آپلود شد، ckeditor به کاربر نشان بده. سپس منطق من به این صورت بوده که مسیری برای ذخیره سازی فایل‌های تغییر اندازه داده شده ، و نیز مسیری برای فایل‌های با اندازه اصلی در نظر گرفتیم. همچنین من در اینجا من از بانک اطلاعاتی برای ذخیره سازی اطلاعاتی از عکس استفاده کردم، که در اینجا بحث اصلی ما نیست. سپس به کمک WebImage.GetImageFromRequest فایل آپلود شده را دریافت کردم. این متد به اندازه کافی باهوش هست که بفهمد ، چه فایلی آپلود شده. سپس پسوند فایل را از نام فایل جدا کردم، و تاریخ کنونی را به شکل رشته در آورده و به انتهای نام عکس اضافه کرده تا از تکراری نبودن نام عکس‌ها مطمئن باشم. سپس پسوند فایل را نیز دوباره به نام فایل اضافه کردم و به کمک متد Save کلاس WebImage عکس را ذخیره کردم. سپس چک کردم که اگر عرض عکس بیشتر از 400 پیکسل هست ، آن را تغییر اندازه بده و ذخیره کنه، و در غیر این صورت آدرس عکسی که قرار بود تغییر اندازه داده بشه با آدرس عکس اصلی یکی میشه.

قسمت مهم:

نکته مهم اینه که ما آدرسهای عکس‌های آپلود شده را چگونه به ckeditor برگردانیم. همان طوری که در قسمت آخر هم مشاهده می‌کنید ، ما سه دستور جاوا اسکریپت به مرورگر برگردوندیم: اولیش:

```

window.parent.CKEDITOR.tools.callFunction(" + CKEditorFuncNum + ", \" + thumbPath + "\", \" + message + \"\");

```

در حقیقت ما در اینجا ما از api های ckeditor و همچنین پارامترهای ارسالی از طرف ckeditor استفاده کردیم ، تا قسمت آدرس عکس را با آدرس عکس تغییر اندازه داده شده و کوچک شده پر می‌کنیم.

سوال؟

حالا چگونه قسمت پیوند را پر کنیم؟ این را دیگر من پیدا نکردم ، تا دست به دامن دوست و یا شایدم دشمن قدیممون جاوا اسکریپت شدم.
اول رفتم به کمک فایرباگ دیدم که id فیلد پیوندها چیه؟

```
<div class="cke_dialog_ui_input_text" role="presentation">
  <input id="cke_145_textInput" class="cke_dialog_ui_input_text"
  type="text" aria-labelledby="cke_146_label">
</div>
```

همان طور که معلومه id این فیلد cke_145_textInput هست و به کمک یه خط js می‌توان این فیلد را با آدرس عکس آپلود شده با سایز اصلی پر کرد.
اولش من این را نوشتم:

```
document.getElementById("cke_145_textInput").value = defaultPath;
```

اما بازم js شروع کرد به بدقلق شدن. هرچی توی کنسول دیباگش کردم خطای null بودن را میداد. بعد از یه ساعت سرو کله زدن و تقریبا ناامید شدن ، چشمم به قسمت اول کد api خود ادیتور که اولش را با window.parent شروع کرده افتاد ، و من هم کد خودم را به شکل زیر تغییر دادم:

```
window.parent.document.getElementById("cke_145_textInput").value = defaultPath;
```

موفقیت در این قسمت از کد باعث شد که من دست به کدتر شوم و مشکل border عکس ها، که به صورت دیفالت در IE یا همون دشمن همیشگی وجود داره ، را حل کنم ومقدار border را به صورت پیش فرض صفر کنم.

```
window.parent.document.getElementById('cke_125_textInput').value=0;
```

خب همه‌ی این دردسرها را ما تحمل کردیم تا به ساده‌ترین شکل ممکن هر عکسی را که آپلود شد ، برای مکانیزم lightbox آماده کنیم و به راحتی یه گالری عکس خوب داشته باشیم.
خب حتما می‌پرسید که از چه پلاگینی برای ایجاد lightbox استفاده کنیم:
من به شخصه پلاگین colorbox را پیشنهاد می‌دم.
با انجام یک سرچ ساده سایتش را پیدا کنید و با مستندات و ویژگی‌های آن آشنا شوید.
یک پیشنهاد:
برای انجام سلکت زدن برای عناصری که باید پلاگین colorbox روی آنها اعمال شوند من سلکت زدن به شیوه‌ی زیر را پیشنهاد می‌کنم:

```
$(".container [href$='.jpg']").colorbox({ maxWidth: 800, opacity: 0.5, rel: 'gal' });
$(".container [href$='.png']").colorbox({ maxWidth: 800, opacity: 0.5, rel: 'gal' });
$(".container [href$='.gif']").colorbox({ maxWidth: 800, opacity: 0.5, rel: 'gal' });
```

فرض کنید div ی که متن و عکس‌های ما را شامل میشه ، کلاسش container باشه . با کمک [href\$='.jpg'] می‌توان گفت هر لینکی که، پسوند فایلی که به آن اشاره می‌کند، jpg هست، ویژگی colorbox را به خود بگیرد.

یک پیشنهاد برای تشکیل گالری عکس:

همان طور که من در بالا اشاره کردم ، rel را بر روی gal قرار دادم، تا هر تگی که ویژگی rel را داشته باشد، تشکیل یک گروه

برای گالری عکس را بدهد.

برای اینکه بتوانیم این ویژگی را به عناصر مورد نظر خود اعمال کنیم ، بازم دست به دامن jQuery می‌شویم:

```
$(".container [href$='.jpg']").attr("rel", "gal");
```

خب مثل اینکه دیگر کار تمام شده و امیدوارم برای شما مفید بوده باشه.
موفق باشید...

نظرات خوانندگان

نویسنده:

علی

تاریخ:

۱۸:۱۶ ۱۳۹۱/۰۵/۲۸

با سلام و تشکر ببخشید می‌خواستم دلیل شما را برای استفاده نکردن از Ckfinder را بپرسم

نویسنده:

بهزاد شیرانی

تاریخ:

۲۰:۲ ۱۳۹۱/۰۹/۲۲

ckfinder مجانی نیست و امکانات زیادی به شما نمیده

نویسنده:

hossein101211

تاریخ:

۱۰:۲۹ ۱۳۹۲/۰۴/۲۳

با سلام

من یه مشکلی دارم اونم اینکه عکس‌ها آپلود میشه و اضافه هم میشه ولی تگ a ساخته نمیشه
امکانش هست کدهای مثال(سورس پروژه) رو بذارید؟

نویسنده:

محسن خان

تاریخ:

۱۱:۵۶ ۱۳۹۲/۰۴/۲۵

[پروژه IRIS](#) از CKEditor هم استفاده کرده. یک نگاهی به سورسش داشته باشید.

نویسنده:

مهدی سعیدی فر

تاریخ:

۱۵:۴ ۱۳۹۲/۰۴/۲۵

سلام

با فایرفاکس تست کنید تا اگه مشکلی داشتید راهنمایی کنم.