

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 textarea با ckeditor 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 ببینیم که این فرم آپلود ckeditor ، چه پارامترهایی را به اکشن متد من ارسال می‌کند ، از فایرباگ کمک گرفتیم:

```
<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 یه 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 apiهای 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 تقریباً 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

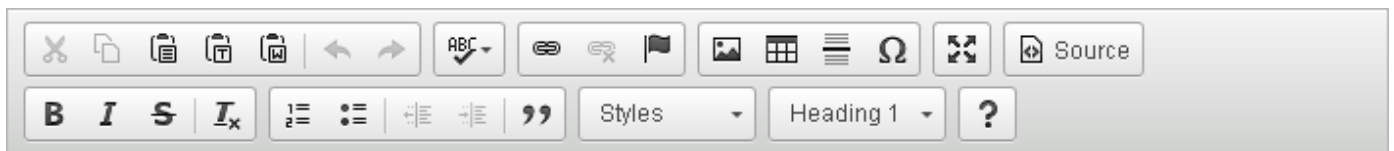
سلام

با فایرفاکس تست کنید تا اگه مشکلی داشتید راهنمایی کنم.

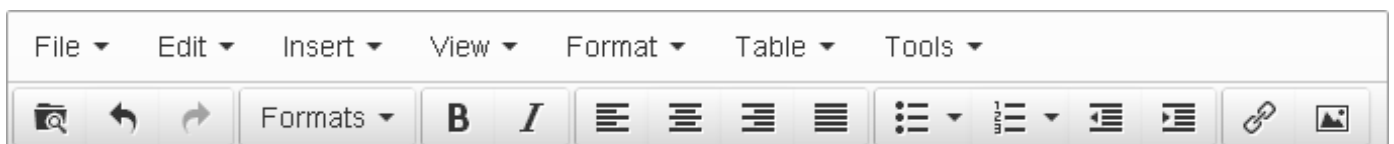
این دو ادیتور یعنی [CKEditor](#) و [TinyMCE](#) دو تا از محبوب‌ترین ادیتورهای موجود تحت وب هستند که به صورت متن باز ارائه می‌شوند و از لحاظ قدرت و کارایی در رده بالایی قرار دارند. همچنین مستندات و apiهای خوبی هم در مورد آنها موجود است؛ ولی در هنگام استفاده خیلی‌ها شاید این سؤال را داشته باشند که کدام ادیتور را انتخاب کنند؟ حتی اگر هر دو ادیتور امکاناتی بیش از نیاز ما را فراهم کنند، باز هم انسان در پی بهترین هاست. در این مطلب به مزایا و معایب اشاره نشده و قصد تخریب ادیتور خاصی را نداریم؛ بلکه خصوصیات آنها بررسی شده و در نهایت توسعه دهنده می‌تواند بر اساس این‌ها، ابزار درستی را برای کارش، انتخاب کند.

### رابط کاربری یا user interface

ckeditor دارای رابط کاربری است که برای همه ادیتورها به صورت پیش فرض وجود دارد و این حالت برای خیلی از کاربرها شناخته شده هست.



در tinymce رابط کاربری به غیر از نوار ابزار میتواند شامل منو هم باشد؛ با اینکه استفاده از منو آنچنان در ادیتور مورد استقبال قرار نمی‌گیرد ولی ممکن است بعضی‌ها این ویژگی را دوست داشته باشند:



برای حذف منو در این ادیتور یا مشخص کردن چه عناصری نمایش داده شود، این کد در سایت سازنده نوشته شده:

```
// Disable all menus
tinymce.init({
  menubar : false
});

// Configure different order
tinymce.init({
  menubar: "tools table format view insert edit"
});
```

### پلاگین‌ها Plugins

وقتی بحث پلاگین پیش کشیده می‌شود، انتخابی بین ساده بودن و اختصاصی بودن پیش خواهد آمد. به نظر می‌رسد پلاگین‌های ck در حالت عادی بیشتر از tiny هستند و امکانات بیشتری را نسبت به tiny ارائه می‌کنند؛ برای مثال به دیالوگ درج تصویر در دو شکل زیر نگاه کنید: در ck شما امکانات بیشتری برای درج یک تصویر دارید تا tiny



Image Properties

Image Info

Link

Upload

URL

Alternative Text

Width

Height

Border

HSpace

VSspace

Alignment

<not set>

Preview

>Lorem ipsum dolor sit amet, consectetur  
adipiscing elit. Maecenas feugiat consequat diam.  
Maecenas metus. Vivamus diam purus, cursus a,  
commodo non, facilisis vitae, nulla. Aenean dictum  
lacinia tortor. Nunc iaculis, nibh non iaculis  
aliquam, orci felis euismod neque, sed ornare  
massa mauris sed velit. Nulla pretium mi et risus.  
Fusce mi pede, tempor id, cursus ac, ullamcorper  
nec, enim. Sed tortor. Curabitur molestie. Duis velit  
augue, condimentum at, ultrices a, luctus ut, orci.  
Donec pellentesque egestas eros. Integer cursus,  
augue in cursus faucibus, eros pede bibendum  
...

OK

Cancel

TinyMCE

در نسخه‌های پیشین tiny این دیالوگ‌ها به صورت پنجره‌های popup باز می‌شدند، ولی در تمامی نسخه‌های ck بسیاری از این پنجره‌ها بدین صورت باز نشده و سفارشی هستند. درست است که در ظاهر مثل هم باز می‌شوند ولی مکانیزم و روش باز شدن آن‌ها با یکدیگر متفاوت است و در واقع ck با روشی هوشمندانه و با استفاده از کدهای جاوا اسکریپت و ترکیب لایه‌های html این روش را پیاده سازی می‌کند.

شاید بگویید خوب چه فرقی می‌کند که از چه روشی برای پیاده سازی استفاده شود؟ این مورد از دو جنبه مورد بررسی قرار می‌گیرد:

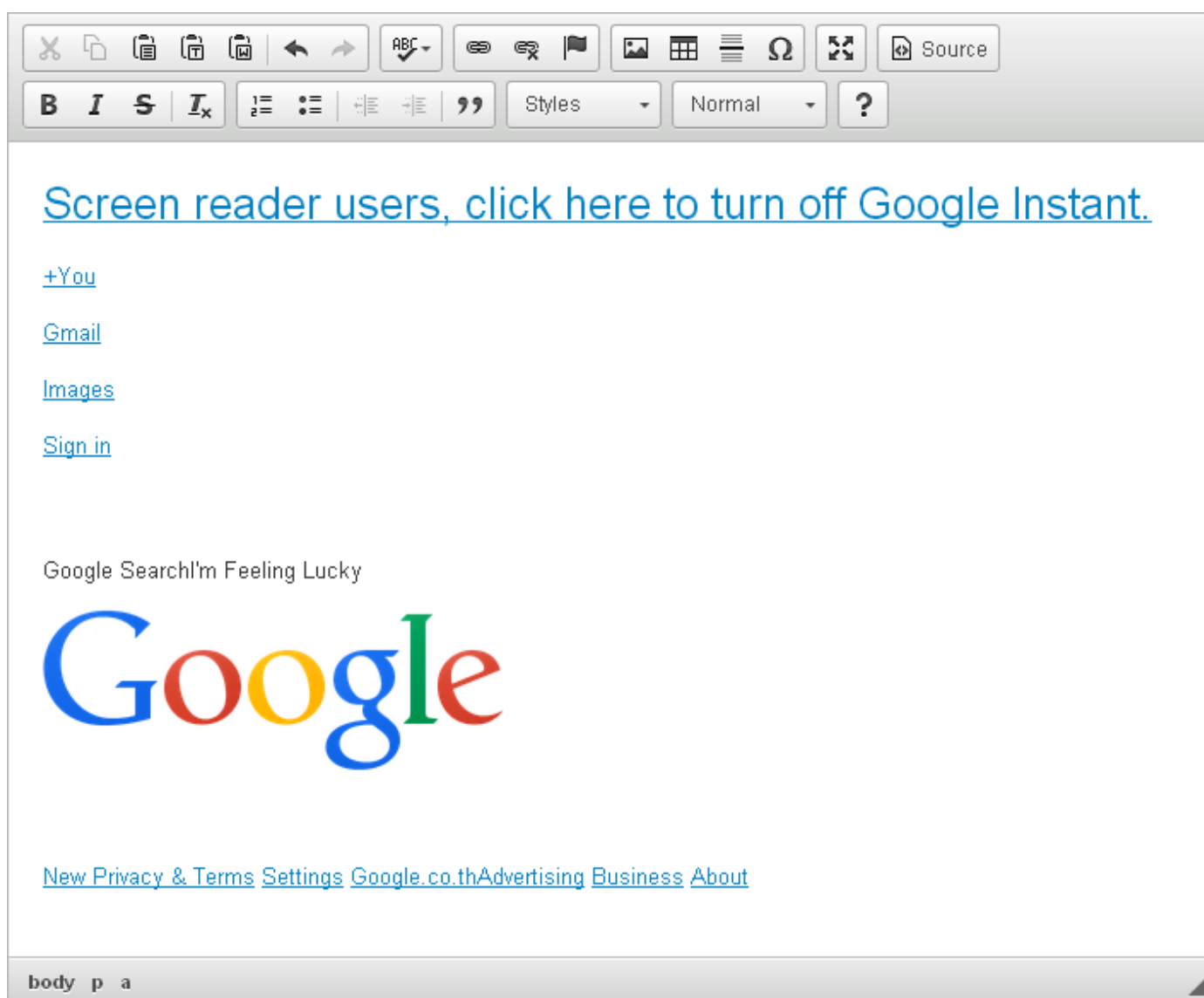
- 1- اگر به مرورگرهایی چون IE دقت کرده باشید، می‌بینید که موقعی که یک popup باز می‌شود، به عنوان یک پنجره‌ی جدید باز می‌شود و باعث شلوغ شدن صفحه می‌گردد و بیشتر موجب آزار کاربر می‌گردد.
- 2- ساخت واقعی یک صفحه popup منجر شدن به ایجاد یک پنجره ای جدید و طی شدن فرآیندهای مربوط به بارگذاری و رندر شدن یک صفحه‌ی جدید می‌باشد؛ ولی در مورد پنجره‌های سفارشی این فرآیندها صورت نمی‌گیرد. این مورد بر روی بسیاری از هاست‌ها خود را به وضوح نمایش می‌دهد و ck در صرفه جویی از زمان، به مراتب بهتر عمل می‌کند.

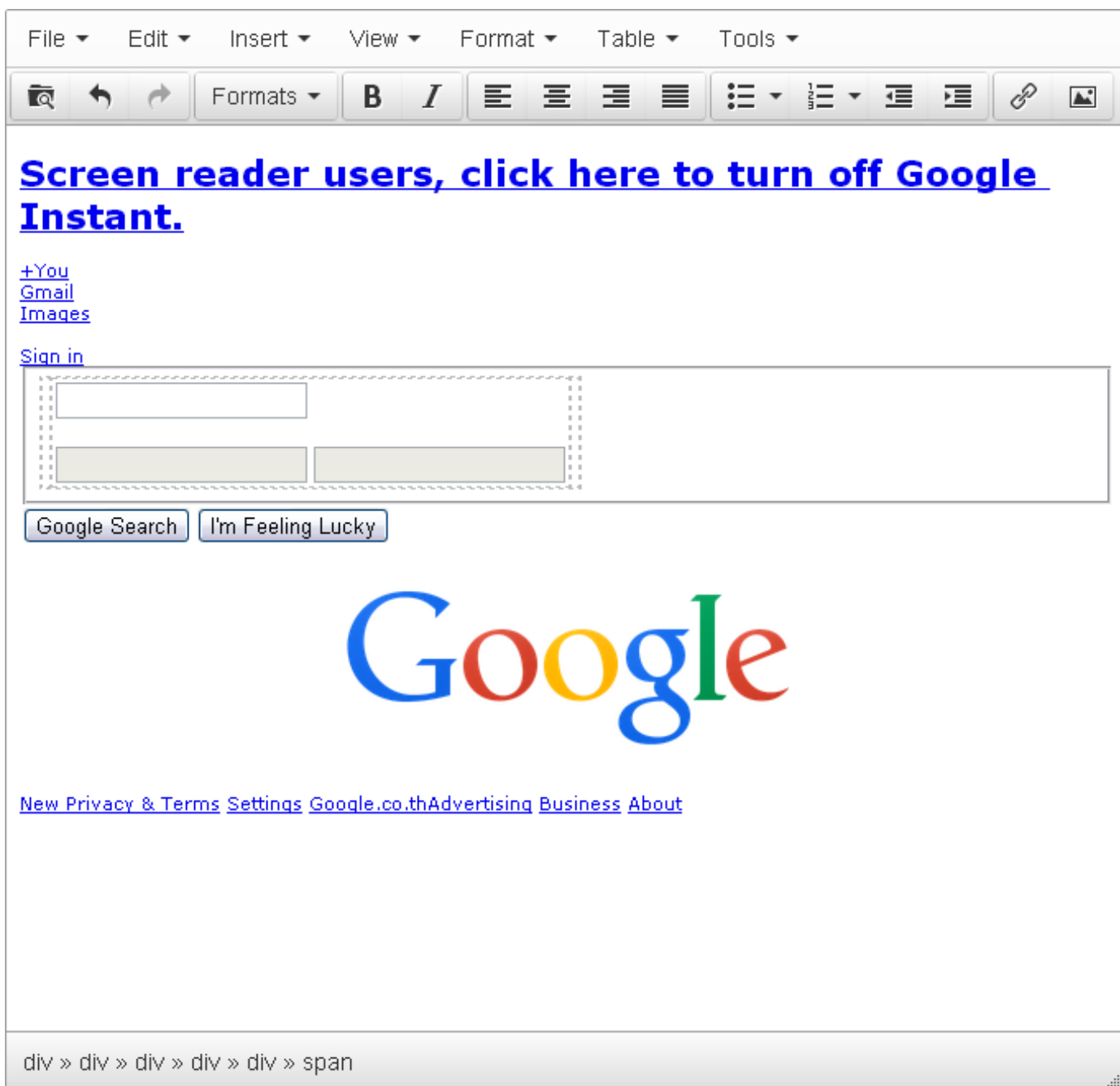
از نسخه 4 tinymce به بعد مکانیزم سفارشی بودن دیالوگ‌ها بر روی این ادیتور نیز صورت گرفته است.

### کپی و پیست Copy - Paste

بسیاری از کاربران گاهی اوقات مطالب خود را از جایی یا از یک صفحه‌ی وب دیگر به ادیتور کپی می‌کنند. برای مثال در اینجا صفحه‌ی گوگل را به داخل ادیتورها کپی می‌کنیم و به نظر میرسد ckeditor در این مواقع رفتار بهتری نسبت به tiny از خود نشان می‌دهد.

در تائینی لوگوی گوگل دقیقاً در وسط قرار گرفته و بعضی موارد خاص مانند فرم‌ها به خوبی نمایش داده می‌شوند؛ ولی در ck شما متن تمیزتری دارید و طوری نشان داده می‌شود که یک css کاربرپسند به آن اضافه شده است.





### انعطاف پذیری Flexibility

هر دو ادیتور در این زمینه به خوبی طراحی شده‌اند و با استفاده از جاوا اسکریپت می‌توان آن‌ها را به سادگی توسعه داد و امکانات دلخواهتان را به آن اضافه کنید ولی در کل ckeditor در زمینه api‌های برنامه نویسی، پیشروتر از رقیبش tinymce است.

### حجم یا سایز ادیتورها Size

با توجه به رشد اینترنت در گوشی‌های همراه و دیگر وسایل همراه، حجم منابع اینترنتی بیش از قبل مورد توجه قرار می‌گیرد. در این زمینه tinymce برگ برنده‌ای داشته و بهتر است بدانید که حجم این ویرایشگر تقریباً نصف رقیبش یعنی ckeditor می‌باشد.

حالا شما با توجه به موارد بالا می‌توانید ادیتور خود را انتخاب کنید.

در اینجا هم می‌توانید فایل‌های مربوط به زبان را نیز دانلود کنید که هر دو از زبان فارسی پشتیبانی می‌کنند.

[فایل‌های زبان TinyMCE](#)

[فایل‌های زبان CKEditor](#)

منابع:

<http://www.krizalys.com/article/ckeditor-vs-tinymce>

<http://www.tinymce.com/wiki.php/Configuration:menubar>

## نظرات خوانندگان

نویسنده: وحید نصیری  
تاریخ: ۹:۱۴ ۱۳۹۳/۰۹/۱۲

قسمت ارسال فایل CKEditor که قبلا به نام FCKEditor شناخته می‌شد، به شدت مورد علاقه‌ی ربات‌های اسکن سایت است:

Page not found: /manage/editor/filemanager/browser/default/connectors/test.html [Details...](#)

Page not found: /js/FCKEditor/editor/filemanager/upload/test.html [Details...](#)

Page not found: /js/FCKEditor/editor/filemanager/connectors/uploadtest.html [Details...](#)

Page not found: /js/fckeditor/editor/filemanager/connectors/test.html [Details...](#)

Page not found: /js/fckeditor/editor/filemanager/browser/default/connectors/test.html [Details...](#)

Page not found: /FCKEditor/editor/filemanager/upload/test.html [Details...](#)

Page not found: /FCKEditor/editor/filemanager/connectors/test.html [Details...](#)

Page not found: /FCKEditor/editor/filemanager/connectors/uploadtest.html [Details...](#)

Page not found: /fckeditor/editor/filemanager/browser/default/connectors/test.html [Details...](#)

Page not found: /fckeditor/editor/filemanager/browser/default/connectors/test.html [Details...](#)

Page not found: /editor/filemanager/browser/default/connectors/test.html [Details...](#)

Page not found: /editor/filemanager/connectors/test.html [Details...](#)

Page not found: /editor/editor/filemanager/connectors/test.html [Details...](#)

Page not found: /editor/editor/filemanager/upload/test.html [Details...](#)

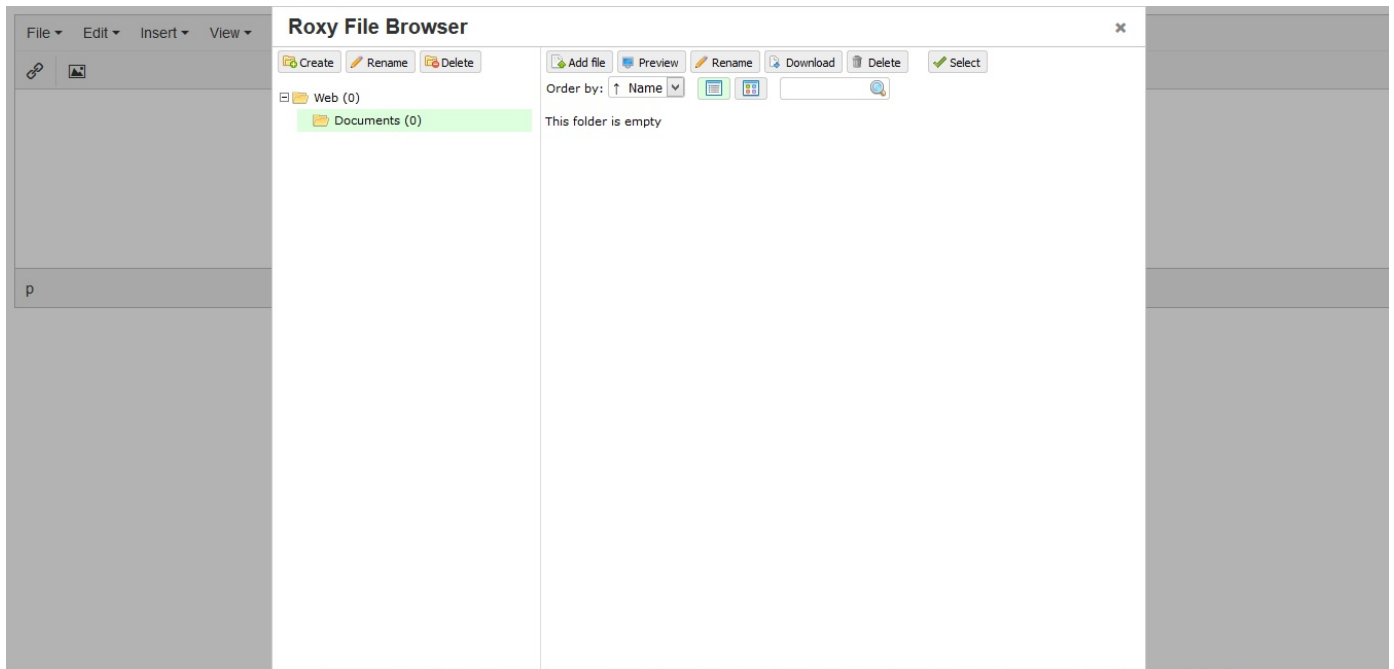
Page not found: /app/agents/agents/fckeditor/editor/filemanager/browser/default/browser.html [Details...](#)

هر روز حداقل یکبار سایت را برای یافتن این مورد اسکن (کور) می‌کنند.

نویسنده: افشار محبی  
تاریخ: ۱۹:۴۱ ۱۳۹۳/۰۹/۱۲

من در چند پروژه از ckeditor استفاده کردم. خیلی راضی هستم ازش. مشکلات راست به چپ و این طور چیزها هم نداره خوشبختانه.

در این مقاله سعی در پیاده سازی [Roxy FileMan](#) در Asp.net mvc داریم. البته خود وب سایت آن ( [اینجا](#) ) برای .net Framework 4.5 پروژه ای را آماده کرده است که بنده بنا دارم همان پروژه را توضیح دهم. قبل از شروع به کار، ابتدا نمایی از این ابزار مدیریت فایل (File Manager) را ملاحظه فرمایید:



1- ابتدا یک پروژه ی جدید MVC می سازیم

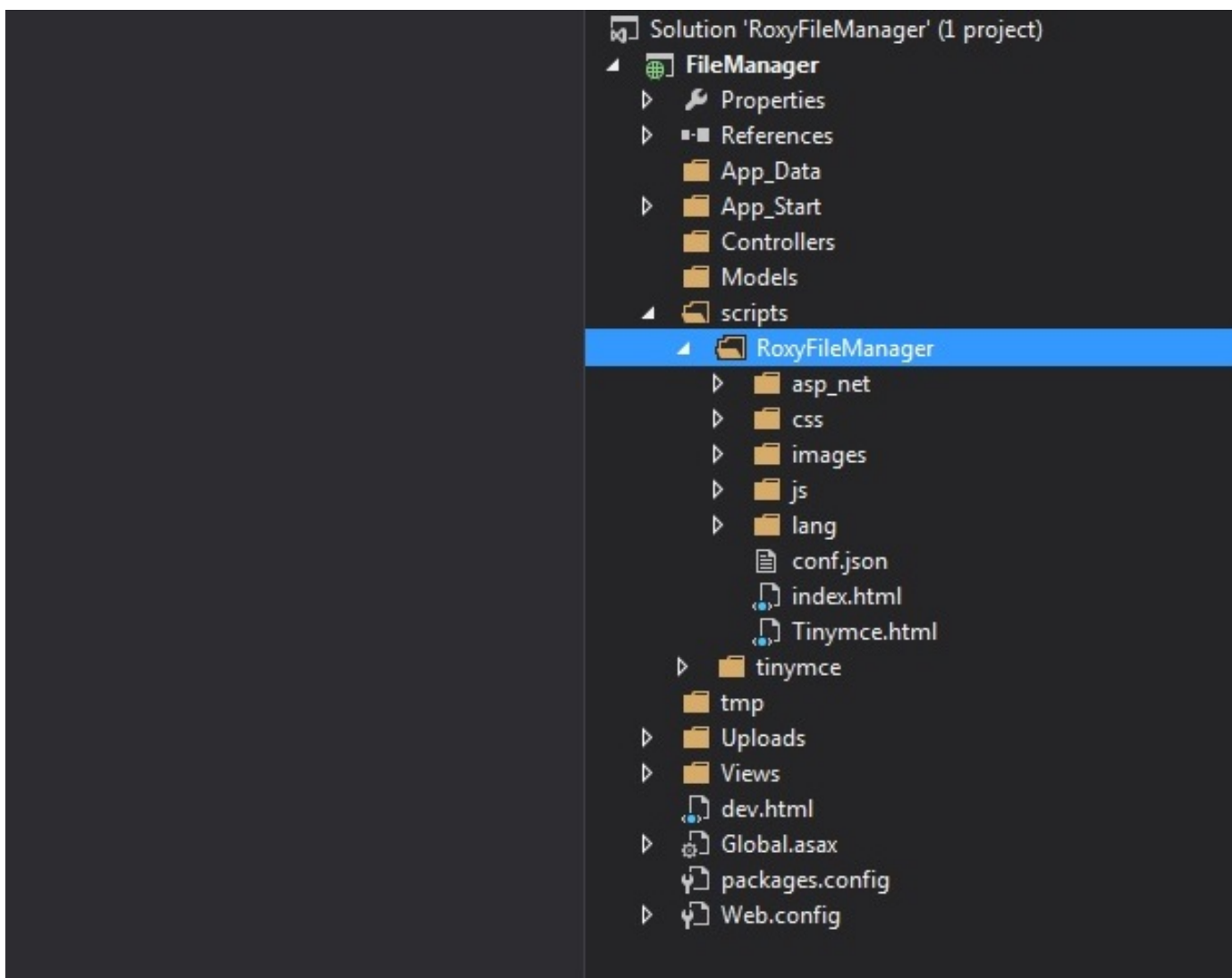
2- جهت بخش سمت سرور، کد هندلری به نام main.ashx نوشته شده است که در فولدر asp\_net قرار دارد و در ادامه به

توضیح آن خواهیم پرداخت. [با هندلر بیشتر آشنا شوید.](#)

3- پیشنهاد می شود جهت راحتی کار و انتقال بین پروژه های خود در فولدر scripts فولدری را به نام RoxyFileManager بسازید.

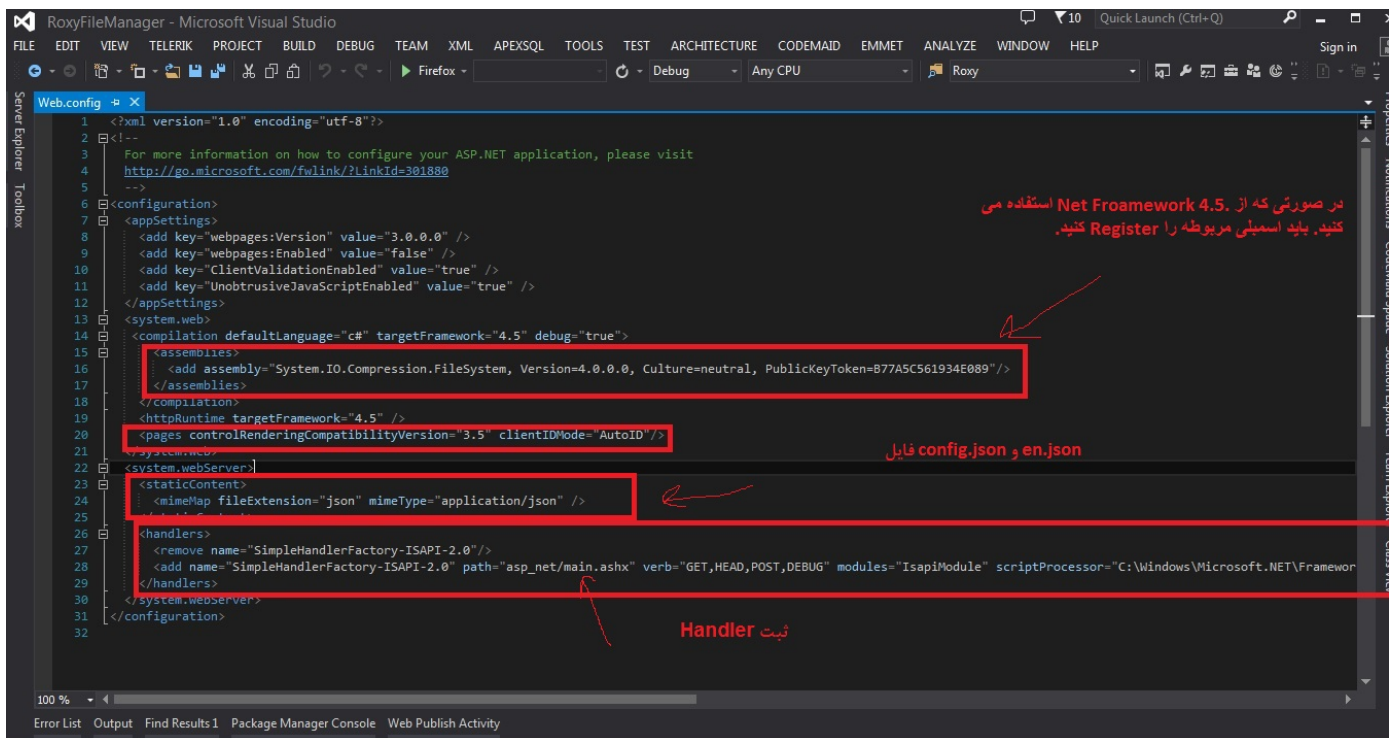
این کار سبب می شود جهت پروژه های مختلف فقط فولدر RoxyFileManager را بین پروژه های خود کپی و پیست کنید.

4- از داخل پروژه دانلود شده، فولدر های lang، js، Images، css، asp\_net را به پروژه ی جدید کپی می کنیم.



5- اکنون به سراغ web.config می‌رویم. موارد قرمز ضروری هستند که به web.config اضافه شوند.





6- کل مدیریت فایل در فایل index.html می باشد و اگر تمایل به تغییر عنوان، درج لوگوی خودتان و ... را دارید به این فایل مراجعه کنید.

7- فایل conf.json، همانطور که از نام این فایل برداشت می شود، تنظیمات را می توان به فرمت json در این فایل جستجو کرد. مهمترین تنظیمات، بحث تغییر زبان می باشد که می توان با تعویض قسمت LANG از en به هر زبان دلخواهی که داخل پوشه ی lang می باشد، آن را تغییر داد که البته بنده آن را به زبان فارسی هم ترجمه کرده ام که در فولدر lang جهت استفاده موجود می باشد.

8- اکنون به فولدر asp\_net می رویم و با handler بیشتر آشنا می شیم. در خط های اول، به متغیر confFile بر می خوریم که آدرس فایل conf.json را داده است.

9- آدرس دایرکتوری اصلی را در متد GetFilesRoot جستجو کنید. (من معمولاً در پروژه های خود، فولدری دارم به نام uploads که داخل آن تمام فولدرها که مربوط به مازول های مختلف است، نگه داری می شوند و خود کاربر آپلود می کند. اکنون برای این پروژه می توان یک فولدر جدید باز کرد و به آن آدرس داد تا در روند پروژه ی شما اختلالی ایجاد نشود. من نام آن فولدر را web گذاشته ام) نکته جالب همین جاست که شما به کمک این آدرس دهی از source پروژه می توانید خیلی کارها را انجام دهید. مثلاً در یک پروژه می خواهید به فولدرها طبق نقشی که کاربران در سیستم دارند دسترسی بدهید تا فایل ها را آپلود کنند و ... به راحتی می توانید با کمی کد نویسی این تغییرات را در متد GetFilesRoot اعمال کنید.

اگر بخواهید مدیریت فایل را بر اساس زبان سیستم تغییر دهید کافیست، یک فایل conf.json دیگر ایجاد کنید و در سورس به ازای هر زبان، آن فایل conf.json را لود کنید. البته در هر فایل شما باید زبان را تغییر دهید. راست چین کردن این ادیتور را در اسرع وقت انجام می دهیم و برای شما آپلود می کنم. تا کاملاً نسخه ی بومی سازی آن در اختیار شما قرار گیرد.

10- اکنون به سراغ تنظیمات Tiny mce می رویم.

خوب، ابتدا با دستور Install-Package TinyMCE در Package Manager Console و انتخاب سورس Nuget آخرین نسخه را نصب می کنیم. در حال حاضر آخرین نسخه نگارنده 4.2.5 می باشد.

کل این تنظیمات در فایل TinyMce.html می باشد (این فایل در پروژه شما می تواند همان view ها و partial view های شما باشد که تنظیمات اولیه tinymce را می نویسد). حال بررسی مواردی که به tinymce اضافه می شوند تا با مدیریت فایل ادغام شود:

## افزونه مدیریت فایل های رایگان Roxy FileMan برای TinyMce و CkEditor

```

1  <script type="text/javascript" src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
2  <script src="/scripts/tinyMce/tinyMce.min.js"></script>
3  <textarea id="tinyMce" name="tinyMce" rows="10" cols="80"></textarea>
4  <script>
5      // This must be set to the absolute path from the site root.
6      var roxyFileman = 'index.html?integration=tinymce4';
7      $(function () {
8          tinymce.init({
9              selector: "textarea",
10             valid_elements: ".*[*]",
11             convert_urls: false,
12             valid_children: "+body[i],+body[section],+div[i]",
13             theme: "modern",
14             height: 300,
15             plugins: [
16                 "advlist autolink link image lists charmap print preview hr anchor pagebreak spellchecker",
17                 "searchreplace wordcount visualblocks visualchars code fullscreen insertdatetime media nonbreaking",
18                 "save table contextmenu directionality emoticons template paste textcolor"
19             ],
20             content_css: "/Plugins/tinyMce/content.css",
21             toolbar: "insertfile-undo-redo | styleselect | bold italic | alignleft aligncenter alignright alignjustify | bullist numlist outdent indent | link image | p
22             file_browser_callback: RoxyFileBrowser,
23             style_formats: [
24                 { title: 'Bold text', inline: 'b' },
25                 { title: 'Red text', inline: 'span', styles: { color: '#ff0000' } },
26                 { title: 'Red header', block: 'h1', styles: { color: '#ff0000' } },
27                 { title: 'Example 1', inline: 'span', classes: 'example1' },
28                 { title: 'Example 2', inline: 'span', classes: 'example2' },
29                 { title: 'Table styles' },
30                 { title: 'Table row 1', selector: 'tr', classes: 'tablerow1' }
31             ]
32         });
33     });
34 
```

و اسکریپتی جهت افزونه‌ی مدیریت فایل Roxy FileMan را در پایین متد اضافه می‌کنیم.

```

39  });
40
41  function RoxyFileBrowser(field_name, url, type, win) {
42      var cmsURL = roxyFileman; // script URL - use an absolute path!
43      if (cmsURL.indexOf("?") < 0) {
44          cmsURL = cmsURL + "?type=" + type;
45      }
46      else {
47          cmsURL = cmsURL + "&type=" + type;
48      }
49      cmsURL += '&input=' + field_name + '&value=' + win.document.getElementById(field_name).value;
50      tinyMCE.activeEditor.windowManager.open({
51          file: cmsURL,
52          title: 'Roxy File Browser',
53          width: 850, // Your dimensions may differ - toy around with them!
54          height: 650,
55          resizable: "yes",
56          plugins: "media",
57          inline: "yes", // This parameter only has an effect if you use the inlinepopups plugin!
58          close_previous: "no"
59      }, {
60          window: win,
61          input: field_name
62      });
63      return false;
64  }
65  </script>
66  </body>
67  </html>
68

```

در خط ۱۲ آدرس دهی شد  
var roxyFileman = 'index.html?integration=tinymce4';

خوب، اکنون فایل TinyMce.html را اجرا کنید و نتیجه را ملاحظه بفرمایید.

Roxy FileMan را می‌توان با CKEditor هم سازگار کرد، که جهت اطلاعات بیشتر [اینجا](#) را مطالعه فرمایید. فایل پروژه:



## نظرات خوانندگان

نویسنده:

میثم 99

تاریخ:

۱۱:۵۲ ۱۳۹۴/۰۷/۰۳

سلام؛ ممنون بابت مطلب خوبتون. برای امنیت سرور که کسی نتونه فایل شل آپلود کنه. چگونه می توان مسیر فایلها رو به App\_Data تغییر داد؟ برای عکس مشکلی نیست ولی باز کردن نوع فایلهای دیگر مثل pdf فیلمها. فقط چک کردن پسوند کفایت نمی کنه. فکر کنم در پوشه App\_Data باشه امنیت بیشتری دارد.

نویسنده:

وحید نصیری

تاریخ:

۱۲:۳۶ ۱۳۹۴/۰۷/۰۳

برای امنیت آپلود، مراجعه کنید به مطالبی مانند:

« [تشخیص نوع فایل با استفاده از محتوای فایل](#) »

« [کنترل عمومی فایل های آپلودی در ASP.NET MVC](#) »

« [محدود کردن کاربرها به آپلود فایل هایی خاص در ASP.NET MVC](#) »

در مورد `accessPolicy=Read` در نظرات مورد آخری بیشتر بحث شده است و در این حالت حتی اگر یک فایل شل هم آپلود شود، قابلیت اجرا نخواهد داشت.