

چند سال قبل یک کنترل آپلود فایل در برنامه‌های ASP.NET Web forms در سایت Code projects [منتشر شد](#) که من در چند پروژه از آن استفاده کردم. در ادامه نحوه سازگار سازی این مجموعه را با ASP.NET MVC مرور خواهیم کرد:

### الف) سورس‌های اصلی Flash کنترل ارسال فایل‌ها

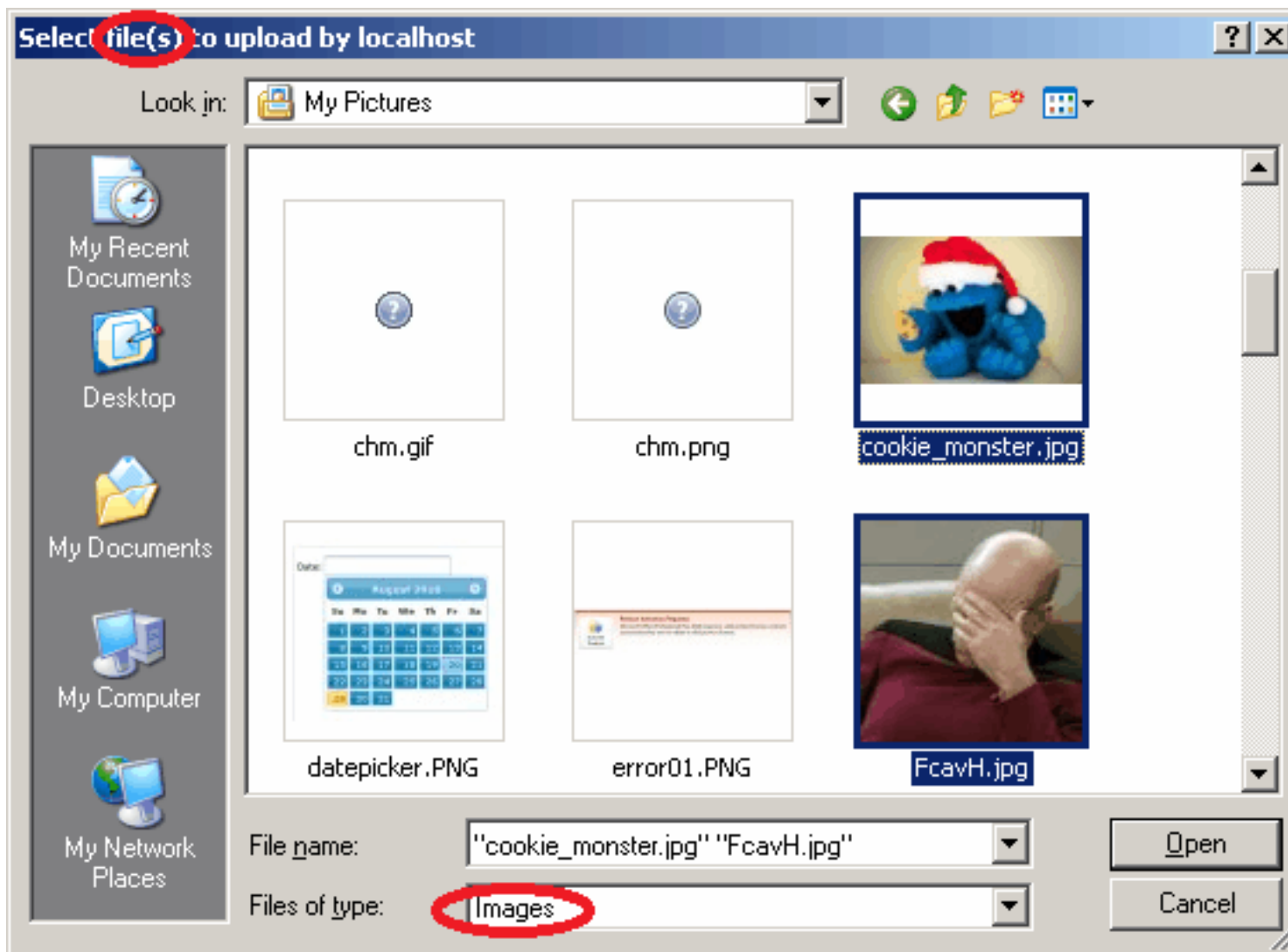
اگر علاقمند به تغییر اطلاعاتی در فایل فلش نهایی هستید به پوشه OriginalFlashSource پروژه پیوست شده مراجعه کنید. در اینجا برای مثال یک سری از برچسب‌های آن فارسی شده‌اند و کامپایل مجدد.

## Flash Uploader

The screenshot displays the Flash Uploader interface. It features two file upload slots. The first slot shows 'FcavH.jpg-1.76KB' with a progress bar at 'Uploaded 0%' and a 'حذف از لیست' (Remove from list) button. The second slot shows 'cookie\_monster.jpg-17.04KB' with a progress bar at 'Uploaded 0%' and a 'حذف از لیست' button. To the right of these slots are three buttons: 'افزودن فایلها' (Add files), 'حذف لیست' (Remove list), and 'ارسال' (Upload). At the bottom, a summary bar indicates 'Uploaded 0% - 0.00 bytes of 18.81KB'.

### ب) مزیت استفاده از Flash uploader

با استفاده از Flash uploader امکان انتخاب چندین فایل با هم وجود دارد. همچنین در صفحه دیالوگ انتخاب فایل‌ها دقیقاً می‌توان پسوند فایل‌های مورد نظر را نیز تعیین کرد. این دو مورد در حالت ارسال معمولی فایل‌ها به سرور و استفاده از امکانات معمولی HTML وجود ندارند. به علاوه امکان نمایش درصد پیشرفت آپلود فایل‌ها و همچنین حذف کلی لیست و حذف یک آیتم از لیست را هم در نظر بگیرید.



ج) معادل کنترل Web forms را در ASP.NET MVC به شکل زیر می‌توان تهیه کرد:

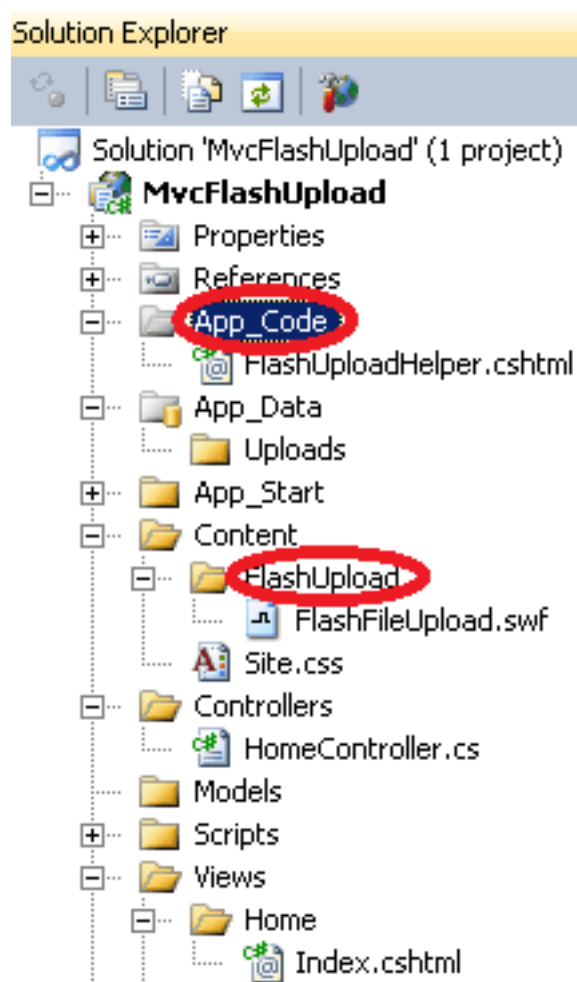
```
@helper AddFlashUploader(
    string uploadUrl,
    string queryParameters,
    string flashUrl,
    int totalUploadSizeLimit = 0,
    int uploadFileSizeLimit = 0,
    string fileTypes = "",
    string fileTypeDescription = "",
    string onUploadComplete = "")
{
    onUploadComplete = string.IsNullOrEmpty(onUploadComplete) ? "" : "completeFunction=" +
onUploadComplete;
    queryParameters = Server.UrlEncode(queryParameters);
    fileTypes = string.IsNullOrEmpty(fileTypes) ? "" : "&fileTypes=" + Server.UrlEncode(fileTypes);
    fileTypeDescription = string.IsNullOrEmpty(fileTypeDescription) ? "" : "&fileTypeDescription="
+ Server.UrlEncode(fileTypeDescription);
    var totalUploadSizeLimitData = totalUploadSizeLimit > 0 ? "&totalUploadSize=" +
totalUploadSizeLimit : "";
    var uploadFileSizeLimitData = uploadFileSizeLimit > 0 ? "&fileSizeLimit=" + uploadFileSizeLimit
: "";
    var flashVars = onUploadComplete + fileTypes + fileTypeDescription + totalUploadSizeLimitData +
uploadFileSizeLimitData + "&uploadPage=" + uploadUrl + "?" + queryParameters;
    <object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0"
width="575" height="375" id="fileUpload" align="middle">
    <param name="allowScriptAccess" value="sameDomain" />
    <param name="movie" value="flashUrl" />
    <param name="flashVars" value="flashVars" />
    <param name="onUploadComplete" value="onUploadComplete" />
    </object>
}
```

```

<param name="movie" value="@flashUrl" />
<param name="quality" value="high" />
<param name="wmode" value="transparent">
<param name="FlashVars" value="@flashVars">
<embed src="@flashUrl"
FlashVars="@flashVars"
quality="high"
wmode="transparent"
width="575"
height="375"
name="fileUpload"
align="middle"
allowScriptAccess="sameDomain"
type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer" />
</object>
}

```

این اطلاعات در فایلی به نام FlashUploadHelper.cshtml در پوشه App\_Code قرار خواهند گرفت.



(د) نحوه استفاده از HTML helper فوق:

```

@{
    ViewBag.Title = "Index";
    var uploadUrl = Url.Action("Uploader", "Home");
    var flashUrl = Url.Content("~/Content/FlashUpload/FlashFileUpload.swf");
}

```

```

<h2>
    Flash Uploader</h2>
<div style="background: #E0EBEF;">
    @FlashUploadHelper.AddFlashUploader(
        uploadUrl: uploadUrl,
        queryParameters: "User=Vahid&Id=تست",
        flashUrl: flashUrl,
        fileTypeDescription: "Images",
        fileTypes: "*.gif; *.png; *.jpg; *.jpeg",
        uploadFileSizeLimit: 0,
        totalUploadSizeLimit: 0,
        onUploadComplete: "alert('انجام شد');")
</div>

```

با کدهای کنترلی معادل:

```

using System.Collections.Generic;
using System.IO;
using System.Web;
using System.Web.Mvc;

namespace MvcFlashUpload.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult Uploader(string User, string Id, IEnumerable<HttpPostedFileBase> FileData)
        {
            var queryParameter1 = User;
            var queryParameter2 = Id;
            // ...

            foreach (var file in FileData)
            {
                if (file.ContentLength > 0)
                {
                    var fileName = Path.GetFileName(file.FileName);
                    var path = Path.Combine(Server.MapPath("~/App_Data/Uploads"), fileName);
                    file.SaveAs(path);
                }
            }

            return Content(" ");
        }
    }
}

```

### توضیحات:

در اینجا uploadUrl، مسیر اکشن متدی است که قرار است اطلاعات فایل‌ها را دریافت کند. queryParameters اختیاری است. اگر تعریف شود تعدادی کوئری استرینگ دلخواه را می‌تواند به متد Uploader ارسال کند. برای نمونه در اینجا User و Id ارسال شده‌اند یا هر نوع کوئری استرینگ دیگری که مدنظر است.

flashUrl مسیر فایل SWF را مشخص می‌کند. در اینجا فایل FlashFileUpload.swf در پوشه Content/FlashUpload قرار گرفته است.

fileTypeDescription برچسبی است که نوع فایل‌های قابل انتخاب را به کاربر نمایش می‌دهد و fileTypes نوع‌های مجاز قابل ارسال را دقیقاً مشخص می‌کند.

پارامترهای uploadFileSizeLimit و totalUploadSizeLimit در صورتیکه مساوی صفر وارد شوند، به معنای عدم محدودیت اندازه در فایل‌ها و جمع حجم ارسالی در هر بار است.

استفاده از پارامتر onUploadComplete اختیاری است. در اینجا می‌توان پس از پایان عملیات از طریق جاوا اسکریپت عملیاتی را انجام داد. برای مثال اگر خواستید کاربر را به صفحه خاصی هدایت کنید، window.location مقدار دهی نمائید.

در متد Uploader کنترلر فوق، پارامترهای User و id اختیاری بوده و بر اساس queryParameters متد

FlashUploadHelper.AddFlashUploader مشخص می‌شوند. اما نام FileData نباید تغییری کند؛ از این لحاظ که دقیقاً همین نام در فایل فلش، مورد استفاده قرار گرفته است. در اکشن متد دریافت فایل‌ها، لیستی از فایل‌های ارسالی به سرور دریافت شده و سپس بر این اساس می‌توان آن‌ها را در مکانی مشخص ذخیره نمود.

#### دریافت پروژه

[MvcFlashUploader.zip](#)

## نظرات خوانندگان

نویسنده: سعید یزدانی  
تاریخ: ۱۳۹۱/۱۱/۲۲ ۱۵:۳۰

سلام؛ اگه میشه یک مقدار بیشتر در مورد پیاده سازی helper توضیح بدید  
با تشکر

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۱۱/۲۲ ۱۷:۵

مراجعه کنید به مبحث «نوشتن HTML Helpers ویژه، به کمک امکانات Razor» [در قسمت هشتم](#) سری MVC برای مشاهده توضیحات تکمیلی این روش.

نویسنده: سعید یزدانی  
تاریخ: ۱۳۹۱/۱۱/۲۲ ۱۷:۱۳

منظورم از پیاده سازی ، بعضی از کدها و تگ هایی هست که استفاده کردید مثل تگ object .

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۱۱/۲۲ ۱۷:۲۸

این تگ object روش متداول اضافه کردن active-x ها در صفحات HTML است. اطلاعات بیشتر : [^](#)

نویسنده: غدیری  
تاریخ: ۱۳۹۱/۱۱/۲۶ ۱۹:۴۵

برای آپلود کردن تصاویر به روشهای بسیار بهتر و سریعتر با استفاده از HTML و jQuery می‌توانید از [jquery.plupload](#) که امکانات بسیاری از جمله

انتخاب چندین فایل با هم

تعیین پسوند فایل‌های مورد نظر

امکان نمایش درصد پیشرفت آپلود فایل‌ها

همچنین حذف کلی لیست

حذف یک آیتم از لیست

و غیره را دارد استفاده فرمایید و نیازی هم به استفاده از فلش نمی‌باشد البته [jquery.plupload](#) بر حسب اینکه Browser کلاینتها عمل می‌کند و به صورت خودکار از نوعی که خود Browser کلاینتها می‌توانند استفاده کنند استفاده می‌نماید برای توضیحات بیشتر [لینک](#) را ببینید.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۱۱/۲۶ ۲۰:۴۱

یک مطلب در موردش قبلا در سایت مطرح شده:

« [آپلود فایل‌ها با استفاده از PlUpload در Asp.Net Mvc](#) »