

تا قبل از آمدن html5 امکان آپلود چندین فایل در Asp.net web forms امکان پذیر نبود و کاربران می‌بایستی فایل‌های مورد نظر خود را یکی یکی انتخاب و آپلود می‌کردند که تا حد زیادی سخت و حوصله زیادی هم می‌خواست. اما با معرفی html5 یک attribute به تگ مربوط به آپلود فایل به اسم AllowMultiple افزوده شد که مقادیر قابل قبول این attribute مقادیر بولی true, false می‌باشند. اگر این attribute به تگ‌های مربوط به آپلود فایل اضافه نشود، به صورت پیش فرض قادر خواهید بود که هر بار فقط یک فایل را برای آپلود انتخاب کنید. با مقدار دهی این Attribute با مقدار true این اجازه به شما داده می‌شود که در هر بار بتوانید چندین فایل را به صورت همزمان آپلود نمایید.

کد زیر نمایشی از چگونگی بکاربردن این attribute در تگ input می‌باشد:

```
<input type="file" multiple="multiple" name="FileUpload1" id="FileUpload1" />
```

این Attribute از نسخه ASP.NET 4.5 Framework به Asp.net Web forms اضافه شده و برنامه نویسان می‌توانند در صورت نیاز از این امکان استفاده کنند. کد زیر، نحوه افزودن این Attribute به تگ FileUpload در ASP.Net Web Forms می‌باشد:

```
<asp:FileUpload runat="server" ID="FileUploadMultiple" AllowMultiple="true" />
```

و اما به چه روشی می‌توانیم فایل یا فایل‌های انتخاب شده را در این حالت آپلود کرد؟ در صورتیکه از این روش در پروژه‌هایتان استفاده کنید فقط کافیست با یک حلقه تمامی کنترل‌های مورد نظر را پیمایش و هر کدام از فایل‌ها را آپلود و ذخیره نمایید.

برای درک بهتر مطلب، پروژه جدیدی در Asp.net web Forms ایجاد کرده و کنترل‌های زیر را به آن اضافه کنید:

```
<asp:FileUpload runat="server" ID="FileUploadMultiple" AllowMultiple="true" />
<asp:Button runat="server" ID="btnUlpad" Text="Upload" OnClick="btnUlpad_Click" />
<asp:Label runat="server" ID="lblMessage"></asp:Label>
```

و در رویداد مربوط به دکمه آپلود، قطعه کد زیر را قرار دهید :

```
int Count = 0;
foreach (var item in FileUploadMultiple.PostedFiles)
{
    string Extension = Path.GetExtension(item.FileName);
    string FileName = new Random().Next(1, 50).ToString()+Extension;
    item.SaveAs(Server.MapPath("~/")+"//File//" +FileName);
    Count++;
}
if (Count == FileUploadMultiple.PostedFiles.Count)
    lblMessage.Text = string.Format("فایل‌های انتخابی با موفقیت آپلود شدند");
else
    lblMessage.Text = string.Format("{1} از {0} فایل با موفقیت آپلود شد", Count,
FileUploadMultiple.PostedFiles.Count);
```

کدهای پروژه : [Upload-multiple.rar](#)

## نظرات خوانندگان

نویسنده: علی قربانی  
تاریخ: ۱۸:۱۷ ۱۳۹۴/۰۴/۰۷

با سلام؛ کد شما فقط برای ذخیره فایل است اگر بخواهیم آدرس و اسم فایل رو در دیتابیس ذخیره کنیم به چه صورت باید عمل کنیم؟

نویسنده: وحید نصیری  
تاریخ: ۱۹:۱۴ ۱۳۹۴/۰۴/۰۷

آدرس نهایی ذخیره شده، همان پارامتر متد SaveAs است. نام هم از item.FileName استخراج می‌شود. این دو مقدار را دارید، مانند روش‌های متداول، ذخیره‌اش کنید.

نویسنده: سید محمد حسین موسوی  
تاریخ: ۱۴:۲۲ ۱۳۹۴/۰۴/۰۸

سلام  
راهی وجود داره که بشه ظاهر فایل آپلود رو هم سفارشی کرد؟

نویسنده: علی یگانه مقدم  
تاریخ: ۱۰:۲۰ ۱۳۹۴/۰۴/۰۹

[استایل و +](#)

نویسنده: عثمان رحیمی  
تاریخ: ۱۴:۱۵ ۱۳۹۴/۰۴/۱۰

راه حل مورد نظر برای این کار گذاشتن یک عکس و یا هر تگ دیگر است و در رویداد کلیک تگ مورد نظر، کلیک FileUpload رو فراخوانی کنید: [^](#)

نویسنده: سید محمد حسین موسوی  
تاریخ: ۱۶:۱۵ ۱۳۹۴/۰۴/۱۷

اگر قرار باشه همین کار و با fileupload خود html انجام بدیم، موقع ذخیره کردن چی باید بنویسیم؟  
چون وقتی می‌نویسیم btnFileUpload.PostedFiles خطا می‌گیره

System.Web.UI.HtmlControls.HtmlInputFile' does not contain a definition for 'PostedFiles' and no extension method 'PostedFiles' accepting a first argument of type 'System.Web.UI.HtmlControls.HtmlInputFile' could be found (are you missing a using directive or an assembly reference?)

نویسنده: عثمان رحیمی  
تاریخ: ۲۰:۷ ۱۳۹۴/۰۴/۱۷

ابتدا تگ input خود را در فرم قرار دهید :

```
<input type="file" multiple="multiple" name="File1" id="File1" runat="server" />
```

و در پشت فرم می‌توانید با استفاده از یک حلقه، همه فایل‌های انتخاب شده را آپلود کنید :

```
for (int i = 1; i <= Request.Files.Count; i++)
{
```

```
        var file = Request.Files[i];  
    }
```