



Upload بر روی سرور

آپلود فایل بر روی سرور از طریق کلاینت، یکی از اعمالی است که برنامه نویسان ASP.NET گاهی با آن مواجه می شوند.

بدین منظور، از 2 کلاس در ASP.NET استفاده می شود.

- 1) **System.Web.UI.HtmlControls.HtmlInputFile**
- 2) **System.Web.HttpPostedFile**

اولین کلاس نمایانگر یک کنترل **HtmlInputFile** است که شامل یک **TextBox** به همراه یک دکمه **Browse** به منظور انتخاب فایل آپلود شونده توسط کاربر می باشد. دومین کلاس، نمایانگر فایلی است که توسط کاربر انتخاب شده. این کلاس از اولین کلاس (**HttpPostedFile**) نمونه سازی یا ایجاد می شود.

کلاس **HtmlInputFile**:

این کلاس دارای 4 خاصیت بوده که به ترتیب در زیر شرح داده شده است:

Accept: لیستی از انواع فایل هایی که کاربر قادر به آپلود آنها بر روی سرور است.

MaxLength: حداکثر تعداد کاراکترهای مجاز مسیر انتخاب فایل.

PostedFile: این خاصیت، یک شی از نوع **System.Web.HttpPostedFile** بر می

گرداند که نمایانگر فایل انتخاب شده توسط کاربر جهت آپلود است.

Width: این خاصیت نمایانگر طول **Text Box** ای است که به وسیله **HtmlInputFile**

نمایش داده می شود.

به منظور استفاده از کنترل **HtmlInputFile** و آپلود فایل، مقدار خاصیت **enctype** فرم باید

برابر با **multipart/form-data** قرار گیرد.

مثال:

```
<form enctype="multipart/form-data" runat="server">  
Select File to Upload :
```

```
<input id="uploadedFile" type="file" runat="server">  
<input type="button" id="upload "  
value="Upload "  
OnServerClick="Upload_Click "
```



```
runat="server"<
</form>
```

کلاس **HttpPostedFile**:

این کلاس نمایانگر فایلی است که توسط کاربر جهت آپلود بر روی سرور انتخاب شده. این کلاس دارای یک متد به نام **SaveAs** می باشد که فایل انتخاب شده را بر روی سرور ذخیره می کند. (در این مقاله از روش دیگری بدین منظور استفاده شده است) این کلاس دارای 4 خاصیت بوده که به ترتیب در زیر شرح داده شده است:

- ContentLength**: حجم فایل آپلود شده بر حسب بایت.
- ContentType**: نوع فایل آپلود شده: مثلاً: "image/gif"
- FileName**: مسیر فایل انتخاب شده توسط کاربر.

نکته مهم: تنها مرورگرهایی که بر روی ویندوز اجرا می شوند، مقدار این خاصیت را بر می گردانند. **InputStream**: یکی شی از نوع **System.IO.Stream** که به فایل آپلود شده اشاره کرده و اجازه کار با فایل را فراهم می کند.

<hr>

نکته مهم: خاصیت **PostedFile** کنترل **HtmlInputFile**، به فایلی که بر روی سرور آپلود شده اشاره می کند. پس در صورتی که عمل آپلود با موفقیت انجام نشده باشد، این خاصیت مقدار **null** را بر می گرداند. به شکل زیر به سادگی می توان از انجام موفقیت آمیز عمل آپلود اطمینان حاصل کرد.

```
If ) HtmlInputFile1.PostedFile != null(
```

```
}
```

```
//Done
```

```
{
```

```
else
```

```
}
```

```
//Fail
```

```
{
```

```
>hr<
```

پس از انتخاب فایل توسط کاربر، باید یک اشاره گر به فایل انتخاب شده ایجاد گردد:



```
HttpPostedFile myFile = HtmlInputFile1.PostedFile;
```

از آنجایی که قصد ذخیره محتویات فایل در یک آرایه (بافر) را داریم، باید اندازه آرایه را برابر با اندازه فایل تعیین کنیم:

```
int nFileLen = myFile.ContentLength;  
byte[] myData = new byte[nFileLen];
```

حال نوبت ریختن محتویات فایل در آرایه با استفاده از متد **Read** خاصیت **InputStream** مرجع شی **HttpPostedFile** است.

```
myFile.InputStream.Read(myData, 0, nFileLen);
```

تنها کار باقی مانده، ذخیره محتویات آرایه بر روی سرور است. تابع زیر این کار را به راحتی با دریافت مسیری که قرار است فایل بر روی آن ذخیره شود و نام آرایه انجام میدهد:

```
private void SaveFile(string strPath, ref byte[] Buffer(  
}
```

```
FileStream myFile = new FileStream(strPath ,  
FileMode.Create);
```

```
myFile.Write(Buffer, 0 ,Buffer.Length);
```

```
myFile.Close();
```

```
{
```

نکته مهم: به منظور آپلود فایل بر روی سرور، اجازه دسترسی **Write Permission** به پوشه ای که قرار است فایل در آن ذخیره شود، الزامی است. موفق باشید.

مرجع : www.barnamenevis.org

نویسنده : **Behruz_Rad**