

الرحيم الرحمن الله بسم

داده تحليل آموزشگاه

ايران در ديتابيس و نويسي برنامه مركز ترين تخصصي

آموزش آپلود

مدرس : مهندس افشين رفوا

آموزش آپلود / بارگذاري فايل در سرويس دهنده با استفاده از تکنولوژی ASP.NET

ASP.NET دو کنترل ویژه ی بارگذاري (آپلود) فايل در سرويس دهنده (**server**) ارائه می دهد. مجرد اينکه سرور داده های فايل ارسال (**post**) شده را دريافت می کند، اپليکيشن مورد نظر می تواند آن را ذخيره کرده، بررسي کند و يا آن را كاملا نادیده بگیرد. دو کنترل زیر امکان بارگذاري فايل را در سرور فراهم می کند.

HtmlInputFile – يك کنترل سرور HTML

FileUpload – یکی از کنترل های وب که متعلق به خود **ASP.NET** می باشد

هر دو کنترل نام برده قابليت بار گذاري فايل در سرور را دارند، با اين تفاوت که **FileUpload** به صورت خودکار رمزگذاري (**encoding**) فرم را تنظيم (**set**) می کند در حالی که کنترل **HtmlInputFile** چنین کاری را انجام نمی دهد.

در اين فصل از سری مقاله های آموزشی **ASP.NET** تنها به شرح **FileUpload** می پردازيم. کنترل ذکر شده به کاربر اجازه می دهد فايلي که قرار است آپلود شود را جستجو (مور) کرده و انتخاب کند. اين کار را با ارئه ی يك دکمه ی پيمائش (**browse button**) و کادر متن (**text box**) برای وارد کردن اسم فايل مورد نظر انجام می دهد.

پس از اینکه کاربر اسم فایل را در کادر متن وارد کرد، متد **SaveAs** کنترل نام برده فراخوانده شده و فایل مورد نظر داخل دیسک ذخیره می شود.

دستور نگارش (**syntax**) کنترل **FileUpload** به صورت زیر می باشد.

```
<asp:FileUpload ID="Uploader" runat="server" />
```

کلاس **FileUpload** از کلاس **WebControl** مشتق شده و از این رو تمامی اعضای آن را به ارث می برد. علاوه بر اعضای به ارث گرفته شده کلاس **FileUpload** دارای خواص **read-only** (خاصیت یا **property** هایی که تنها دارای **getter** هستند **read-only** اطلاق می گردند) زیر می باشد.

شرح	خاصیت
خاصیت FileBytes آرایه ای از بایت های موجود در یک فایل که قرار است در سرور بارگذاری (upload) شود را برمی گرداند.	FileBytes
این خاصیت یک شی Stream باز می گرداند که به فایلی که قرار است با استفاده از کنترل FileUpload در سرور بارگذاری شود اشاره می کند.	FileContent
اسم فایلی که قرار است آپلود شود را برمی گرداند.	FileName
این خاصیت مشخص می کند آیا کنترل مورد نظر فایلی برای ارسال و بارگذاری در سرور دارد یا خیر.	HasFile
ارجاعی (reference) به فایل آپلود شده بازمی گرداند.	PostedFile

فایل ارسال (**post**) شده در شی از نوع **HttpPostedFile** کپسوله سازی می شود. می توان به وسیله ی خاصیت **PostedFile** کلاس **FileUpload** به فایل ارسال شده دسترسی پیدا کرد.

کلاس **HttpPostedFile** دارای خواص بسیار پرکاربرد زیر است.

شرح	خاصیت
اندازه یا حجم فایل بارگذاری شده در سرور پس دهنده را بر حسب واحد بایت برمی گرداند.	ContentLength

ContentType	نوع (type) MIME فایل آپلود شده در سرور را برمی گرداند.
FileName	این خاصیت اسم کامل فایل بار گذاری شده در سرور را باز می گرداند.
InputStream	یک شی stream که به فایل آپلود شده اشاره می کند باز می گرداند.

مثال

مثال زیر کنترل **FileUpload** و خاصیت های آن را نمایش می دهد.

فرم مورد نظر دارای یک کنترل **FileUpload**، دکمه ی ذخیره سازی (**save button**) و یک کنترل **label** ویژه ی نمایش اسم فایل، نوع و طول آن می باشد.



کد **Content file** به ترتیب زیر می باشد.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="File_Uploading.aspx.cs"
Inherits="Asp_Net_File_Uploading.File_Uploading" %>
```

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>File_Uploading</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<h3>File Upload:</h3>
<br />
```

```

<asp:FileUpload ID="FileUpload1" runat="server" />
<br />
<br />
<asp:Button ID="btnsave" runat="server" OnClick="btnsave_Click" Text="Save" Style="width:
85px" />
<br />
<br />
<asp:Label ID="lblmessage" runat="server" />
</div>
</form>
</body>
</html>

```

Code behind دکمه ذخیره سازی (**save button**) نیز به شرح زیر می باشد.

```

protected void btnsave_Click(object sender, EventArgs e)
{
    StringBuilder sb = new StringBuilder();
    if (FileUpload1.HasFile)
    {
        try
        {
            sb.AppendFormat(" Uploading file: {0}", FileUpload1.FileName);
            //saving the file
            FileUpload1.SaveAs("<c:\\SaveDirectory>" + FileUpload1.FileName);
            //Showing the file information
            sb.AppendFormat("<br/> Save As: {0}", FileUpload1.PostedFile.FileName);
            sb.AppendFormat("<br/> File type: {0}", FileUpload1.PostedFile.ContentType);
            sb.AppendFormat("<br/> File length: {0}", FileUpload1.PostedFile.ContentLength);
            sb.AppendFormat("<br/> File name: {0}", FileUpload1.PostedFile.FileName);
        }
        catch (Exception ex)
        {
            sb.Append("<br/> Error <br/>");
            sb.AppendFormat("Unable to save file <br/> {0}", ex.Message);
        }
    }
    else
    {
        lblmessage.Text = sb.ToString();
    }
}

```

به نکات زیر دقت کنید.

کلاس **StringBuilder** از فضای نام **System.IO (namespace)** مشتق شده و از این رو حتما باید لحاظ شود.

بلوک های **try** و **catch** به منظور ضبط و نمایش پیام های خطا بکار می روند.