

基于 ASP.NET 的文件上传模块的设计与实现

袁芳

(江西财经职业学院信息工程系, 九江 332000)

摘要: 文件上传是 Web 程序开发中常用的功能, 有着广泛的应用。该文分析了 ASP.NET 中几种常用的文件上传技术, 由于利用 ASP.NET 内置的上传组件来上传文件有所限制, 重点阐述了同时多个文件上传的技术, 并给出了解决方案。

关键词: 上传; 大文件; 多文件; ASP.NET

中图分类号: TP393.093

文献标识码: A

DOI: 10.3969/j.issn.1003-6970.2014.07.032

本文著录格式: [1] 袁芳. 基于 ASP.NET 的文件上传模块的设计与实现 [J]. 软件, 2014, 35 (7): 135-139

1 引言

近年来, 随着互联网高速发展, web 技术发展非常迅猛并且逐渐成熟, 其中, 在 web 技术的应用中文件上传与下载技术是一个很实用的技术, 主要应用在论坛、邮箱、办公网以及其它信息交流比较频繁的网络应用中。在以前的 Web 应用程序中, 要实现文件上传是个很麻烦的事, 但在 ASP.NET 中要实现文件的上传不再是难事。ASP.NET 是微软公司推出新型体系结构 .NET 的一部分, 作为 .NetFrameWork 中重要组成部分的 ASP.NET 已经在内部建立了文件上传的组件, 并且这种功能用起来很方便^{[1][2][3]}。下面介绍一下如何利用 ASP.NET 实现单个和同时多个文件上传技术。

2 单个文件上传

在 ASP.NET 中如果要将文件上传到服务器就要使用到 FileUpload 文件上传控件, 该控件的功能非常强大, 使用起来也非常简单。编写短短几行代码, 就可以实现将选择的文件上传到服务器中。

2.1 FileUpload 控件的使用

FileUpload 控件的主要功能是向制定目录上传文件, 该控件包括一个文本框和一个浏览按钮。用户可以在文本框中输入完整的文件路径, 或者通过按钮浏览并选择需要上传的文件。但 FileUpload 控件不会自动上传文件, 必须设置相关的事件处理程序, 并在程序中实现文件上传, 一般用 Button 控件实现文件上传的事件处理, 设计如图 1 所示。



图 1 单个文件上传设计

在具体应用中, 在单击上传按钮之前先要检查上传的文件是否为空, 这就要用到 FileUpload 控件的属性 HasFile 来检查 FileUpload 控件是否包含要上传的文件, 如果 HasFile 值为 True, 则说明 FileUpload 控件中包含要上传的文件, 则用 FileUpload 控件的 SaveAs 方法把客户端文件上传到服务器端指定目录下。具体代码如下:

```
string savePath = @"c:\temp\uploads\"; // 把文件上传到" c:\temp\uploads\" 目录下
if (FileUpload1.HasFile) // 判断上传文件控件有没有选好文件
{
    string fileName = FileUpload1.FileName; // 获取上传文件的文件名
    savePath += fileName;
    FileUpload1.SaveAs(savePath);
    Response.Write("<script>alert( '你的文件已成功上传!' )</script>");
}
```

```
else
```

```
Response.Write(“<script>alert(‘你没有选好上传文件!’)</script>”);
```

2.2 文件类型检查

在木马病毒横行的今天，我们不能不考虑上传的文件是否符合上传的要求，比如文件类型和文件大小的限制，从而提高服务器整个系统的安全性。因此，我们可以限制用户上传一些特定的文件，上传前必须要进行必要的文件类型检查。比如只能上传 word 文件或 Excel 文件，这样就可以防止一些非法文件上传。下面给出只可以上传 word 文件或 Excel 文件的代码如下：

```
if (FileUpload1.HasFile)
{
String fileName= FileUpload1.FileName; // 获取上传文件的文件名
String extension = System.IO.Path.GetExtension(fileName); // 获取上传文件的扩展名
if ((extension == “.doc”) || (extension == “.xls”)) // 判断扩展名是否是 word 或 Excel 类型
{ savePath += FileUpload1.FileName ;
FileUpload1.SaveAs(savePath);
}
```

2.3 文件大小检查

上传的文件的大小也是要注意的地方。利用 FileUpload 控件的 PostedFile.ContentLength 属性可以判断文件的大小，比如上传文件的大小限制为 2 MB 以下，具体代码如下：

```
if (FileUpload1.HasFile)
{
int fileSize = FileUpload1.PostedFile.ContentLength ; // 获取上传文件的大小
if (fileSize <2097152) // 文件在计算机内部的基本单位是字节，2M=2097152 字节
{ savePath += FileUpload1.FileName ;
FileUpload1.SaveAs(savePath);
}
```

利用 ASP.NET 控件中的 FileUpload 控件时，有时候需要上传大容量的文件，可是默认情况下，上传文件的最大容量为 4MB。如何实现大文件的上传呢？只要在配置文件（Web.Config）中改变两个默认设置即可：httpRuntime 下的 MaxRequestLength 和 requestLengthDiskThreshold, 前者规定了上传的最大容量值，后者设定缓存的大小。

```
<system.web>
...
<httpRuntime maxRequestLength=” 上传文件最大值” requestLengthDiskThreshold=” 缓存大小” />
...
</system.web>
```

3 多个文件同时上传

一般上传文件数为一个文件，但在有些场合要求上传的文件个数很可能是随机的，也就是同时上传多个文件即批量上传，那么我们在设计中就需要通过“添加文件”按钮来添加 FileUpload 控件如图 2 所示，在 FileUpload 控件中选择要上传文件的路径，再循环将 FileUpload 控件中所指定的文件路径通过 SaveAs 方法将上传文件保存在服务

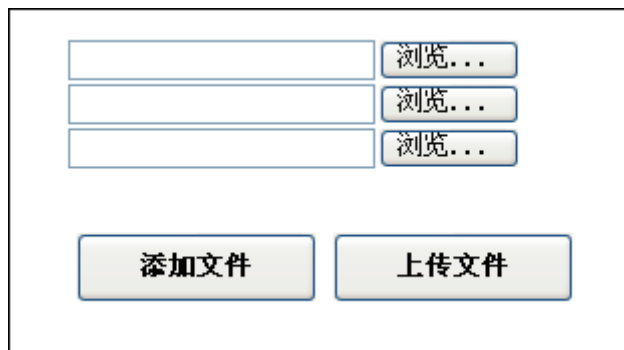


图 2 多个文件同时上传设计界面

器指定文件夹中^[4]。

3.1 动态添加 FileUpload 控件

动态添加 FileUpload 控件可以通过 HtmlTableRow 对象和 HtmlTableCell 对象编辑服务器上的 HTML 的 table 元素来实现。HtmlTableRow 对象表示 HtmlTable 控件中的 tr 元素，HtmlTableCell 对象表示 HtmlTableRow 对象中的 td 和 th 元素，代码如下所示。

```
HtmlTableRow HTR = new HtmlTableRow(); // 表示 HtmlTable 控件中的 <tr> HTML 元素
```

```
HtmlTableCell HTC = new HtmlTableCell(); // 表示 HtmlTableRow 对象中的 <td> 和 <th> HTML 元素
```

```
HTC.Controls.Add((new FileUpload()); // 在单元格中添加一个 FileUpload 控件
```

```
HTR.Controls.Add(HTC); // 在行中添加单元格
```

文件上传通过 HttpFileCollection 对象获取客户端上传文件的集合，循环此集合使用 HttpPostedFile 对象来获取对单个文件的操作^[5]。

3.2 多文件上传功能实现

在页面中拖一个 Table 表格控件，建一个一行一列的表格，在该表格内拖一个 FileUpload 控件，表格外部拖两个 Button 控件实现“添加文件”和“上传文件”的功能，和一个 Label 控件用于显示文件上传是否成功的提示信息。

动态添加控件主要通过 SFUPC（）、InsertC（）和 GetInfo（）三个自定义方法来实现。SFUPC 自定义方法主要用来将文件上传控件集保存在 Session 中，其代码如下所示。

```
private void SFUPC()
{
    ArrayList AL = new ArrayList(); // 创建动态增加数组
    foreach (Control C in F.Controls)
    {
        // 在表格中查找出 FileUpload 控件添加到 ArrayList 中
        if (C.GetType().ToString() == "System.Web.UI.HtmlControls.HtmlTableRow")
        {
            HtmlTableCell HTC = (HtmlTableCell)C.Controls[0];
            foreach (Control FUC in HTC.Controls)
            {
                if (FUC.GetType().ToString() == "System.Web.UI.WebControls.FileUpload")
                {
                    FileUpload FU = (FileUpload)FUC;
                    AL.Add(FU); // 添加 FileUpload 控件
                }
            }
        }
    }
    Session.Add("FilesControls", AL); // 把 ArrayList 添加到 Session 中
}
```

GetInfo（）方法是将保存在 Session 中的文件上传控件集添加到表格中，其代码如下。

```
private void GetInfo()
{
    ArrayList AL = new ArrayList();
```

```

if (Session[ "FilesControls" ] != null)
{
    AL = (ArrayList)Session[ "FilesControls" ];
    for (int i = 0; i < AL.Count; i++)
    {
        HtmlTableRow HTR = new HtmlTableRow();
        HtmlTableCell HTC = new HtmlTableCell();
        HTC.Controls.Add((System.Web.UI.WebControls.FileUpload)AL[i]);
        HTR.Controls.Add(HTC);
        F.Rows.Add(HTR);
    }
}
}

```

InsertC () 方法用来添加一个新的文件上传控件，其代码如下所示。

```

private void InsertC()
{
    ArrayList AL = new ArrayList();
    this.F.Rows.Clear(); // 清除 id 为 F 表格里的所有行
    GetInfo();
    HtmlTableRow HTR = new HtmlTableRow();
    HtmlTableCell HTC = new HtmlTableCell();
    HTC.Controls.Add(new FileUpload());
    HTR.Controls.Add(HTC);
    F.Rows.Add(HTR);
    SFUPC();
}

```

在页面中，通过单击“添加文件” Button 按钮来动态添加文件上传控件，其代码如下。

```

protected void btnAdd_Click(object sender, EventArgs e)
{
    InsertC(); // 执行添加控件方法
    lblMessage.Text = " " ;
}

```

上传文件时通过“上传文件”按钮的 Click 事件将文件上传到 Files 文件夹中，此事件的代码如下。

```

protected void btnUp_Click(object sender, EventArgs e)
{
    if (this.fileup.PostedFile.FileName != " ")
    {
        string FilePath = Server.MapPath( "/" ) + "File" ;
        HttpFileCollection HFC = Request.Files;
        for (int i = 0; i < HFC.Count; i++)
        {

```

```

HttpPostedFile UserHPF = HFC[i]; // 访问指定的文件
try
{
    if (UserHPF.ContentLength > 0)
    {
        // 将上传的文件存储在指定目录下
        UserHPF.SaveAs(FilePath + "\\ " + System.IO.Path.GetFileName(UserHPF.FileName));
    }
}
catch
{
    lblMessage.Text = "上传失败！";
}
}
if (Session[ "FilesControls" ] != null)
{
    Session.Remove( "FilesControls" );
}
lblMessage.Text = "上传成功！";
SFUPC();
}
else
{
    Response.Write( "<script>alert( '上传文件不能为空！' );location=Default.aspx</script>" );
}
}

```

4 小结

文件上传功能在基于 B/S 模式的软件系统中应用非常广泛，是 Web 编程中非常重要的一个技术^[6]，文中主要从单个文件和多个文件上传方面进行了讨论，同时对多个文件同时上传给出了具体的解决方案和代码，该方法简单实用，能轻松完成同时多个文件上传的处理。

参考文献

- [1] 白金牛. ASP.NET 下动态产生控件实现多文件随意上传 [J]. 计算机系统应用, 2007,10
- [2] 兰洋, 蒋顺业. 基于 asp.net 的 FMS 生产管理系统 [J]. 软件, 2013,34
- [3] 王念桥. ASP.NET MVC 模型绑定应用研究 [J]. 软件, 2013,34 (12): 167-169
- [4] 郑伟. ASP.NET 程序设计与软件项目实训 [M]. 清华大学出版社, 2010,1
- [5] 吕双, 房大伟, 刘云峰. ASP.NET 学习手册 [M]. 电子工业出版社, 2011,3
- [6] 徐亚峰, 蒋晓峰. 基于 .NET 平台的交互高校信息系统的设计与实现 [J]. 软件, 2013, 34 (2): 7-9