

Загрузка файлов на сайт по протоколу HTTP на примере depositfiles.com

В .NET Framework нет очевидных средств для загрузки файлов на сайты по протоколу HTTP, разве что WebClient, который, к сожалению, не всегда подходит под стоящие задачи. В этой статье рассмотрен метод загрузки файлов на сайт на примере файлового обменника depositfiles.com с использованием языка C#.

Введение

Для загрузки файлов по протоколу HTTP необходимо знать, как работает этот самый протокол. Нет, вовсе необязательно идти читать умные книжки, придумывать не менее умные поисковые запросы для Google, и даже читать спецификацию протокола HTTP (<http://www.w3.org/Protocols/rfc2616/rfc2616.html>). Хотя конечно, неплохо было бы все это сделать, так, для общего развития. Чтобы понять, как оно работает, достаточно открыть Fiddler и посмотреть, что происходит при отправке файла на сайт.

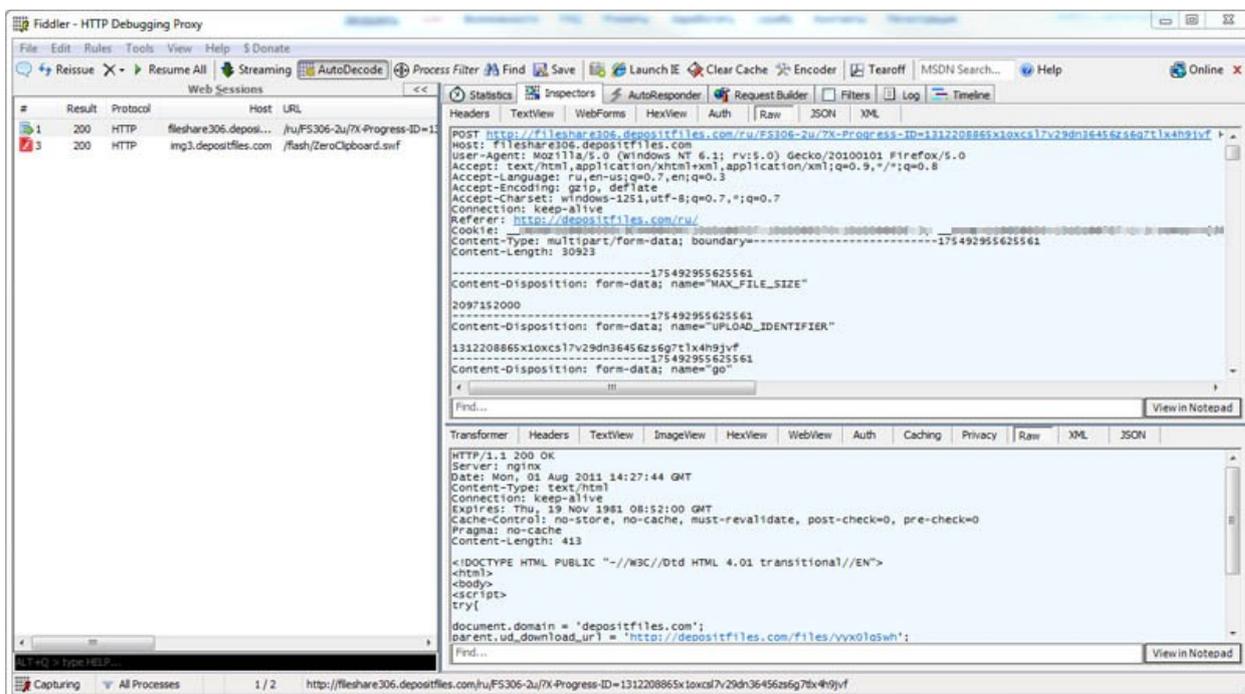


Рис. 1. Просмотр журнала в Fiddler после отправки файла на сервер depositfiles.com.

Как видно на рисунке выше (а еще лучше, если вы будете смотреть прямо в свой Fiddler), отправка файла делается методом **POST**. В поле «**Content-Type**» стоит «**multipart/form-data**» - это важно, т.к. помимо файла в запросе отправляются и другие параметры. После типа содержимого указывается граница – «**boundary**». Граница – это строка, содержащая случайный набор символов. Каждый отправляемый параметр, включая файл, разделяется границей, а также имеет свои заголовки, в частности у текстовых параметров указывается описание содержимого – «**Content-Disposition**», которое всегда равно «**form-data**», а также имя параметра – «**name**». У файла, помимо описания и имени, указывается имя файла – «**filename**», и тип содержимого – «**Content-Type**». Между значением параметра и заголовками параметра всегда идет пара символов перевода строк (CRLF+CRLF). Все параметры передаются «как есть», т.е. не кодируются.

С HTTP вроде бы все понятно, по крайней мере я на это надеюсь, если нет, то форум ждет. Теперь посмотрим на то, как работает depositfiles.com. Если вдруг вы не знаете, что такое depositfiles.com или у вас злобный провайдер запретил доступ к этому сайту, то depositfiles.com – это обычный файлообменник. При загрузке файла, сервисы по обмену файлами обычно работают в два этапа. При первом этапе происходит определение сервера, который готов принять файл. Как правило, у файлообменников может быть множество разных серверов для хранения файлов. Ну а на втором этапе происходит загрузка файла на сервер. И depositfiles.com в этом плане, к нашему счастью, совсем не блещет оригинальностью.

Depositfiles.com имеет некую API, которая позволяет определять «язык» на котором сервер будет выдавать ответы. По умолчанию, при загрузке файлов через веб-интерфейс, depositfiles.com выдает ответы в формате HTML. Но в нашем случае лишняя обработка HTML только усложнит задачу. Поэтому мы будем просить у сервера ответы в формате XML, для этого достаточно указать при запросе параметр format=xml. В других же случаях, скорее всего, придется парсить HTML, хотя этого не так страшно, как может показаться на первый взгляд.

Загрузка

Как я уже говорил, загрузка файлов в depositfiles.com производится в два этапа. Первый этап – запрос адреса сервера, на который можно загрузить файл. На момент написания этой статьи, запрос адреса доступного сервера делался на странице http://depositfiles.com/api/get_upload_info.php Если перейти по этой ссылке, то там будут данные в формате XML. Так что, программно можно сразу загрузить данные в **XmlDocument** методом **Load**, благо этот метод понимает url в качестве имени файла.

```
XmlDocument myXml = new XmlDocument();  
myXml.Load("http://depositfiles.com/api/get_upload_info.php");
```

Примечание. Класс XmlDocument принадлежит пространству имен System.Xml и для его использования необходимо импортировать указанное пространство в проект (using System.Xml).

При успешном выполнении запроса, ссылка для загрузки файла будет находиться в элементе **result/http_upload_path**.

```
string uploadUrl = myXml.SelectSingleNode("result/http_upload_path").InnerText;
```

Имея на руках, а точнее в переменной **uploadUrl**, адрес страницы, которая с нетерпением ждет, чтобы в нее загрузили какой-нибудь файл, да побольше, можно приступать к самой главной части статьи.

Загрузка файла на сервер будет делаться методом POST при помощи класса HttpRequest.

```
HttpRequest myReq = (HttpRequest)HttpRequest.Create(uploadUrl);  
myReq.Method = "POST";
```

Примечание. Для использования класса HttpRequest необходимо импортировать в проект пространство имен System.Net (using System.Net).

Мы будем отправлять на сервер содержимое **multipart/form-data**, для которого нужно сделать границу, чтобы иметь возможность разделять параметры в HTTP-запросе. Границу можно сделать из текущего тика (Ticks) в шестнадцатеричном формате.

```
string boundary = "-----" + DateTime.Now.Ticks.ToString("x");
myReq.ContentType = String.Format("multipart/form-data; boundary={0}", boundary);
```

Нам нужно, чтобы сервер отвечал в формате XML, для этого добавим в запрос параметр «**format**», который будет иметь значение «**xml**».

```
Stream myReqStream = myReq.GetRequestStream()
string header = String.Format("\r\n--{0}\r\nContent-Disposition: form-data; name=\"{1}\"\\
\r\n\r\n", boundary, "format");
byte[] h = System.Text.Encoding.UTF8.GetBytes(header);
myReqStream.Write(h, 0, h.Length);
byte[] b = System.Text.Encoding.UTF8.GetBytes("xml");
myReqStream.Write(b, 0, b.Length);
```

Второй параметр, который мы отправим на сервер, будет сам файл. Сервер depositfiles.com ожидает его в параметре с именем «**files**». В моем случае, файл будет выбран из диалогового окна **OpenFileDialog**.

```
OpenFileDialog ofd = new OpenFileDialog();
ofd.ShowDialog()
header = String.Format("\r\n--{0}\r\nContent-Disposition: form-data; name=\"{1}\";
filename=\"{2}\"\\r\nContent-Type: {3}\\r\n\r\n", boundary, "files",
Path.GetFileName(ofd.FileName), uploadFileType);
h = System.Text.Encoding.UTF8.GetBytes(header);
myReqStream.Write(h, 0, h.Length);
using (FileStream myFile = new FileStream(ofd.FileName, FileMode.Open, FileAccess.Read))
{
    using (BinaryReader myReader = new BinaryReader(myFile))
    {
        byte[] buffer = myReader.ReadBytes(2048);
        while (buffer.Length > 0)
        {
            myReqStream.Write(buffer, 0, buffer.Length);
            buffer = myReader.ReadBytes(2048);
        }
    }
}
```

Примечание. OpenFileDialog является частью сборки System.Windows.Forms.dll и находится в одноименном пространстве. Если вы используете проект типа Console, вы можете подключить сборку System.Windows.Forms.dll к проекту и импортировать пространство имен System.Windows.Forms, а также добавить STAThreadAttribute к главному методу приложения. Классы работы с файлами, включая: Path, FileStream, BinaryReader, находятся в пространстве имен System.IO, чтобы иметь возможность их использовать, импортируйте это пространство имен в проект (using System.IO).

И завершающий штрих – граница в конце запроса.

```
string footer = String.Format("\r\n--{0}--\r\n", boundary);
byte[] f = System.Text.Encoding.UTF8.GetBytes(footer);
myReqStream.Write(f, 0, f.Length);
```

При формировании запроса важно не потерять символы CRLF, ибо это неотъемлемая часть протокола HTTP.

При помощи класса **HttpWebResponse** выполним запрос и загрузим ответ в наш XmlDocument, который был создан в самом начале (при запросе свободного сервера).

```
HttpWebResponse myResp = (HttpWebResponse)myReq.GetResponse();  
StreamReader myStream = new StreamReader(myResp.GetResponseStream(), Encoding.UTF8);  
string str = myStream.ReadToEnd();  
myXml.LoadXml(str);
```

Получить ссылку на загруженный файл можно в элементе **result/download_url**.

```
Process.Start(myXml.SelectSingleNode("result/download_url").InnerText);
```

Примечание. Класс Process принадлежит пространству имен System.Diagnostics, необходимо импортировать его проект (using System.Diagnostics), чтобы иметь возможность использовать этот класс.

Послесловие

Как видите, отправка файлов довольно несложная операция. В этой статье я рассмотрел загрузку файлов на сайт depositfiles.com, однако вы можете использовать приведенные примеры и для других сайтов. Для этого достаточно поменять имя параметра, в котором передается файл. Также может потребоваться добавление других параметров, по аналогии с параметром «format», все зависит от сайта, на который будет производиться загрузка файла. Собственно, в одном запросе файлов может быть и несколько. Как бы там ни было, если у вас возникнут какие либо проблемы или вопросы, пишите на форум.

Скачать пример к статье

--

Алексей Немиро
01 августа 2011 года