
Subject: Re: [FeatureRequests]Use HttpRequest to upload large file

Posted by [kasome](#) on Mon, 14 Jul 2014 08:08:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

hi,

Inorder to upload large file, I make some modification to HttpRequest to do so.

Here is the usage:

```
Upp::int64 uploadFileSourceStartPosition = 1000;
Upp::String uploadFileSourcePath = "TestVideo.mkv";
_httpRequest.ClearPost().PUT().SSL( true ).Url( hostName ).Path( hostPath
).ClearHeaders().Header( "Content-Length", contentLength ).KeepAlive( true ).ContentType(
contentType ).PostDataStream( uploadFileSourcePath, uploadFileSourceStartPosition
).Execute();
//_httpRequest.ClearPost().PUT().SSL( true ).Url( hostName ).Path( hostPath
).ClearHeaders().Header( "Content-Length", contentLength ).KeepAlive( true ).ContentType(
contentType ).PostData( Upp::LoadFile(uploadFileSourcePath).Mid(uploadFileSourceStar
tPosition) ).Execute();
```

The total modification is as follows:

1. upp\uppsrc\Core\Inet.h

Original (version:7373)

```
class HttpRequest : public TcpSocket {
public:
.....
HttpRequest& PostData(const String& pd)          { postdata = pd; return *this; }
.....
HttpRequest& ClearPost()                        { PostData(Null); multipart.Clear(); GET(); return
*this; }
.....
};
```

Modified

```
class HttpRequest : public TcpSocket {
.....
int64      postdataPos; // Added
Upp::String postdataName; // Added
.....
bool  SendingStream(); // Added
```

public:

```
.....
HttpRequest& postData(const String& pd)          { postdata = pd; return *this; }
HttpRequest& postDataStream(const String& pdn, const int64 pos = 0)    { postdataName =
pdn; postdataPos = pos; return *this; } // Added

.....
// HttpRequest& ClearPost()                      { postData(Null); multipart.Clear(); GET(); return
*this; } // Delete
HttpRequest& ClearPost()                        { postData(Null); postDataStream(Null);
multipart.Clear(); GET(); return *this; } // Added

.....
};
```

2. uppsrc\Core\Http.cpp

Original (version:7373)

```
bool HttpRequest::Do()
{
    .....
    case REQUEST:
        if(SendingData())
            break;
        StartPhase(HEADER);
        break;
    .....
}

void HttpRequest::StartRequest()
{
    .....
    if(method == METHOD_GET || method == METHOD_HEAD)
        pd.Clear();
    .....
    if(pd.GetCount() || method == METHOD_POST || method == METHOD_PUT)
        data << "Content-Length: " << pd.GetCount() << "\r\n";
    .....
}

bool HttpRequest::SendingData()
{
    for(;;) {
        int n = min(2048, data.GetLength() - (int)count);
        n = TcpSocket::Put(~data + count, n);
```

```

if(n == 0)
    break;
count += n;
}
return count < data.GetLength();
}

```

Modified

```

bool HttpRequest::Do()
{

```

```

.....
case REQUEST:
    if( IsNull(postdataName) ) {
        if(SendingData())
            break;
    }
    else {
        if(SendingStream())
            break;
    }
    StartPhase(HEADER);
    break;
.....
}

```

```

void HttpRequest::StartRequest()
{

```

```

.....
if(method == METHOD_GET || method == METHOD_HEAD){
    pd.Clear();
    postdataName.Clear();
}
.....
    if((IsNull(postdataName)? pd.GetCount() : Upp::GetFileLength(postdataName)-postdataPos) ||
method == METHOD_POST || method == METHOD_PUT)
    data << "Content-Length: " << (IsNull(postdataName)? pd.GetCount() :
Upp::GetFileLength(postdataName)-postdataPos) << "\r\n";
.....
}

```

```

bool HttpRequest::SendingData()
{

```

```

    for(;;) {
        int n = min(2048, data.GetLength() - (int)count);
        n = TcpSocket::Put(~data + count, n);

```

```

    if(n == 0)
        break;
    count += n;
}
return count < data.GetLength();
}

bool HttpRequest::SendingStream() {
    Upp::FileIn in( postdataName );
    in.Seek( postdataPos );

    int bufferSize = 2048 - (data.GetLength() & 2047);

    Upp::StringBuffer buffer( bufferSize );
    int readingSize = in.Get( buffer, bufferSize );
    int64 postdataSize = in.GetSize() - postdataPos - readingSize;

    data << Upp::String( ~buffer, readingSize );

    for(;;) {
        int n = min(2048, data.GetLength() - (int)count);
        if( data.GetLength() - (int)count < 0 ) {
            int kk = 0;
        }
        n = TcpSocket::Put(~data + count, n);
        if(n == 0)
            break;
        count += n;
    }

    if( data.GetLength() == count ) {
        Upp::StringBuffer buffer( 2048 );
        for(;;) {
            int n = min((int64)2048, (int64)data.GetLength() + postdataSize - (int64)count);
            int readingSize = in.Get( buffer, n );
            n = TcpSocket::Put(~buffer, readingSize);
            if(n == 0)
                break;
            count += n;
        }
    }
    return count < data.GetLength() + postdataSize;
}

```

Hope that helps.

File Attachments

1) [HttpRequest.txt](#), downloaded 402 times
