Subject: Proposition: Adding a way to send huge file via stream in skylark
Posted by Xemuth on Sun, 26 Sep 2021 20:18:52 GMT
View Forum Message <> Reply to Message

While testing my skylark application on a raspberry I noticed that trying to read huge file (upper to 1Go) make the crash app. Indeed the way I did it was the following :

```
http.SetHeader("content-disposition", "attachment; filename=\"" + f->FileName
+"\"").ContentType("application/octet-stream") <<  LoadFile(f->FilePath);
```

Loading a huge file into memory in order to send it in one chunk is not a good idea, especially on low memory platform. That's why I worked on adding a way for skylark to send file in chunk :

```
http.SetHeader("content-disposition", "attachment; filename=\"" + f->FileName
+"\"").ContentType("application/octet-stream").SendFile(f->FilePath);
```

Http.h

```
class Http : public Renderer {
...
private:
FileIn responseStream;
int chunkSize;
...
}
```

Http.cpp

```
Http& Http::SendFile(const Upp::String& filePath, int _chunkSize){
 if(FileExists(filePath)){
  chunkSize = _chunkSize;
  responseStream.Open(filePath);
 }
 return *this;
}
```

Dispatch.cpp

```
void Http::Finalize(bool closeSocket)
{
    if(rsocket) {
        SKYLARKLOG("=== Response: " << code << ' ' << code_text);
```

```
    String r;

    // weird apache2 mod_scgi behaviour
    if(hdr.scgi)
        r << "Status: ";
    else
        r << "HTTP/1.1 ";

    if(redirect.GetCount()) {
        // for SCGI (at least on apache 2 mod_scgi), we need protocol inside url
        if(hdr.scgi && redirect.Find(":") < 0)
            redirect = "http:" + redirect;

        SKYLARKLOG("Redirect to: " << redirect);
        r << code << " Found\r\n";
        r << "Location: " << redirect << "\r\n";
        for(int i = 0; i < cookies.GetCount(); i++)
            r << cookies[i];
    }
    else {
        r <<
            code << ' ' << code_text << "\r\n"
            "Date: " <<  WwwFormat(GetUtcTime()) << "\r\n";
            if(responseStream){
                r << "Content-Length: " << responseStream.GetSize() << "\r\n";
            }else{
  r << "Content-Length: " << response.GetCount() << "\r\n";
            }
            r << "Content-Type: " << content_type << "\r\n";

        for(int i = 0; i < headers.GetCount(); i++)
            r << headers.GetKey(i) << ": " << headers[i] << "\r\n";
        for(int i = 0; i < cookies.GetCount(); i++)
            r << cookies[i];
    }
    r << "\r\n";
    rsocket->PutAll(r);
    if(responseStream){
        Upp::String fileData = "";
while((fileData = responseStream.Get(chunkSize)).GetCount() > 0){
 rsocket->PutAll(fileData);
 fileData = "";
}
rsocket->PutAll("\r\n");
responseStream.Close();
    }else{
        rsocket->PutAll(response);
    }
```

```
        rsocket = NULL;
    }
}
```

Via this new method huge file can be send over a skylark handler

---