Archived:How to upload a file to server with multipart/form-data

Aquivado: Este artigo foi arquivado, pois o conteúdo não é mais considerado relevante para se criar soluções comerciais atuais. Se você achar que este artigo ainda é importante, inclua o template {{ForArchiveReview|*escreva a sua justificativa*}}.

All PySymbian articles have been archived. PySymbian is no longer maintained by Nokia and is not guaranteed to work on more recent Symbian devices. It is not possible to submit apps to Nokia Store.

The example below shows how to upload an image.

First of all, you need to have the **HTTPFileUploader** library installed.

HTTPFileUploader source (HTTPFileUploader.py):

```
## HTTP file uploader
class HTTPFileUploader:
   ## Constructor
   def __init__(self, host, port=80):
      from httplib import HTTP
      self.host = host
      self.port = port
      self.user = None
      self.password = None
      self.page = "upload/uploader.php"
      self.fields = {}
      self.setTargetPath('.')
      self.http = HTTP(self.host, self.port)
      self._result = None

   ## Set user and password
   def setLogin(self, user, password):
      self.user = user
      self.password = password

   ## Select the upload page
   def setPage(self, selector = 'upload/uploader.php'):
      self.page = selector

   ## Add fields in http
   def setField(self, name, value):
      self.fields[name] = value

   ## Change the target path
   def setTargetPath(self, path):
      self.fields["target_path"] = path

   ## Upload file on http server
   def uploadFile(self, aSourceFilename, aKey="file"):
      from base64 import encodestring

      files = ((aKey,aSourceFilename.lower(),open(aSourceFilename,"rb").read()),)
      content_type, body = self.encode_multipart_formdata(self.fields, files)
      self.http.putrequest('POST', self.page)
      if self.user != None and self.password != None:
```


(C) Copyright Nokia 2014. All rights reserved.

```
         self.http.putheader("AUTHORIZATION", "Basic " + \
             encodestring("%s:%s" % (self.user, self.password)).replace("\012", ""))
      self.http.putheader('Content-Type', content_type)
      self.http.putheader('Content-Length', str(len(body)))
      self.http.endheaders()
      self.http.send(body)
      errcode, errmsg, headers = self.http.getreply()
      self._result = self.http.file.read()
      # could need some modification to get the answer: here I just need
      # to get the 5 first characters
      if self._result.strip()[0:5] == "True":
         return True
      else:
         return False

   ## Encode the form.
   #  @param self The object pointer.
   #  @param fields Sequence of (name, value) elements for regular form fields.
   #  @param files Sequence of (name, filename, value) elements for data to be
   #  uploaded as files.
   #  @return (content_type, body) ready for httplib.HTTP instance
   def encode_multipart_formdata(self, fields, files):
      BOUNDARY = '----------ThIs_Is_tHe_bouNdaRY_$'
      CRLF = '\r\n'
      L = []
      for key, value in fields.items():
         L.append('--' + BOUNDARY)
         L.append('Content-Disposition: form-data; name="%s"' % key)
         L.append('')
         L.append(value)
      for file in files:
         key = file[0]
         filename = file[1]
         if len(file) > 2:
            value = file[2]
         else:
            value = None
         L.append('--' + BOUNDARY)
         L.append('Content-Disposition: form-data; name="%s"; filename="%s"' % (key,
filename))
         L.append('Content-Type: %s' % self.get_content_type(filename))
         L.append('Content-Transfer-Encoding: binary')
         if value:
            L.append('')
            L.append(value)
         else:
            L.append('')
            fp = open(filename, 'rb')
            L.append(fp.read())
            fp.close()
      L.append('--' + BOUNDARY + '--')
      L.append('')
      body = CRLF.join(L)
      content_type = 'multipart/form-data; boundary=%s' % BOUNDARY
      return content_type, body

   ## Get the file content type
```


(C) Copyright Nokia 2014. All rights reserved.

```
   def get_content_type(self, filename):
      import mimetypes
      return mimetypes.guess_type(filename)[0] or 'application/octet-stream'

   ## Get result from server
   def getResult(self):
      return self._result

## HTTPS file uploader
class HTTPSFileUploader(HTTPFileUploader):
   ## The constructor
   def __init__(self, host, port=443):
      from httplib import HTTPS
      HTTPFileUploader.__init__(self, host, port)
      self.http = HTTPS(self.host, self.port)
```

Client source:

```
import os
from HTTPFileUploader import * # uploader class

# path of the file to download
filePath = "001.jpg"

# new HTTPFileUploader instance
uploader = HTTPFileUploader('yourServer.xxx', port = 80) # port optional if 80

# set page
uploader.setPage('/uploaderFolder/file_uploader.php')

# Add fields in HTTP: ex file name for or past example
uploader.setField("fileName", os.path.split(filePath)[1])

# upload file - returns True or False.
if not uploader.uploadFile(filePath, "picture"):
    print uploader.getResult()
```

Server source (file_uploader.php):

```
<?php
// In this example a directory "images" needs to be present on the same directory where
// image_uploader.php is, with the necessary rights for the script to write data inside
$content_dir = 'images/';
$filename = Null;

if(isset($_POST['fileName'])){
    $filename = $_POST['fileName'];
}

if(isset($_FILES['picture'])){
    if ($filename == Null){
        $filename = $_FILES['picture']['name'];
    }
```


(C) Copyright Nokia 2014. All rights reserved.

```
    if( !move_uploaded_file($_FILES['picture']['tmp_name'], $content_dir . $filename) ){
        exit("Couldn't write the file in $content_dir");
    }
}
else{
    // something when wrong
    exit("False");
}
// return
echo "True";
?>
```


(C) Copyright Nokia 2014. All rights reserved.