

1.6 Web上传服务器程序

深圳信息职业技术学院 Shenzhen Institute Of Information Technology

教师:黄锐军

日录 COMPANY

1.6.1

Web上传服务器程序

1.6.2

Web上传客户端程序

PART ONE

Web上传服务器程序

Web上传服务器程序



```
服务器程序server.py设计如下:
import flask
app=flask.Flask(__name__)

if __name__=="__main__":
    app.run()
```

```
@app.route("/upload",methods=["POST"])
def uploadFile():
  msg=""
 try:
    if "fileName" in flask.request.values:
      fileName = flask.request.values.get("fileName")
      data=flask.request.get_data()
      fobj=open("upload "+fileName,"wb")
      fobj.write(data)
      fobj.close()
      msg="OK"
    else:
      msg="没有按要求上传文件"
  except Exception as err:
    print(err)
    msg=str(err)
  return msg
```



服务器通过下列方法获取文件名称:

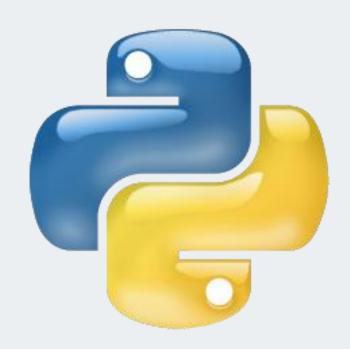
fileName =

flask.request.values.get("fileName")

同时使用flask.request的get_data()函数读取二进制数据:

data= flask.request.get_data()

程序运行后网址为http://127.0.0.1:5000,等待客户端上传文件。如果有文件上传,就读取上传文件的名称与数据,把它存储在磁盘中。



PART TWO

Web上传客户端程序

Web上传客户端程序



客户端要上传二进制数据,要设置表头content-type的值为'application/octet-stream',设置时定义headers字典来指定content-type的值:

headers = {'content-type': 'application/octet-stream'}

这个表头要加入到http请求中,可以先定义一个urllib.request.Request的对象如下:req=urllib.request.Request(url,data,headers)

这个对象的第一个参数url是访问的网址,第二个参数data是要上传的数据,第三个参数headers就是表头字典。



这个程序先不带参数访问http://127.0.0.1:5000,服务器会返回一个可以下载的文件名称给客户端,客户端获取这个文件名称fileName,之后再次采用把文件名称放在地址栏后面的方式再次访问这个网站,即:

data=urllib.request.urlopen(url+"?fileName="+urllib.parse.quote(fileName)) 服务器就把该问价你的二进制数据发送给客户端,客户端接收后就保存到本地,名称是 "download "+fileName。

我们在服务器所在的文件夹中放一个名称为"图像.jpg"的文件,先执行服务器程序建立web网站,然后运行客户端,可以看到下载文件的过程,例如:

准备下载:图像.jpg

下载完毕: 20963 字节

实际上在urllib.request中专门有一个urlretrieve函数是从服务器获取文件保存到本地的,

使用方法是:

urllib.request.urlretrieve(url,localFile)



```
客户端程序client.py设计如下:
import urllib.request
import os
url="http://127.0.0.1:5000/upload"
fileName=input("Enter the file:")
if os.path.exists(fileName):
  fobj=open(fileName,"rb")
  data=fobj.read()
  fobj.close()
  p=fileName.rfind("\\")
  fileName=fileName[p+1:]
  print("准备上传:"+fileName)
  headers = {'content-type': 'application/octet-stream'}
```



```
purl=url+"?fileName="+urllib.parse.quote(fileName)
  req = urllib.request.Request(purl, data, headers)
  msg=urllib.request.urlopen(req)
  msg=msg.read().decode()
  if msg=="OK":
    print("成功上传:",len(data),"字节")
  else:
    print(msg)
else:
  print("文件不存在!")
```

THANK YOU