

---

## 1.6 Web 上传文件

上传文件就是客户端把文件发送给服务器的过程，我们设计一个客户端程序，它能读取本地的一个文件数据，把数据发送给服务器，服务器读取数据，把数据写到文件中，完成上传过程。

### 1.6.1 上传二进制数据

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。

然后通过 `req` 对象调用 `urlopen` 函数就可以向 `url` 服务器传输二进制数据 `data` 了：

```
msg=urllib.request.urlopen(req)
```

在实际调用时还可以把文件名称放在 `url` 后面，同时上传文件名称与数据。

例如：

```
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)
```

### 1.6.2 服务器程序

服务器程序 `server.py` 设计如下：

```
import flask
```

```
app=flask.Flask(__name__)
```

```
@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

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

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。

### 1.6.3 客户端程序

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!")
```

我们先运行服务器程序，然后运行客户端程序，输入要上传的文件的路径与名称，就可以完成上传。