

1.4 POST 方法向网站发送数据

1.4.1 客户端 POST 发送数据

POST 方法访问网站时客户端向服务器发送表单数据，表单数据的组织方式与 GET 方法的参数列表十分相似，结构如下：

“名称 1=值 1&名称 2=值 2&名称 3=值 3……”

多个数据之间用“&”符号隔开，如果参数值包含汉字，那么我们必须使用 `urllib.parse.quote` 对参数值进行编码，例如：

```
province= urllib.parse.quote("广东")
city= urllib.parse.quote("深圳")
data="province="+province+"&city="+city
data=data.encode()
urllib.request.urlopen("http://127.0.0.1:5000",data=data)
```

这里 `data=data.encode()` 是把 `data` 字符串按 utf-8 的编码转为二进制数据。POST 方法与 GET 方法最大的不同是 GET 的参数放在地址栏的后面，而 POST 的数据放在 `urlopen` 函数的 `data` 参数中，而且这个参数值必须是二进制数据。

编程客户端 `client.py` 程序如下：

```
import urllib.parse
import urllib.request
url="http://127.0.0.1:5000"
try:
    province= urllib.parse.quote("广东")
    city= urllib.parse.quote("深圳")
    data="province="+province+"&city="+city
    data=data.encode()
    html=urllib.request.urlopen("http://127.0.0.1:5000",data=data)
    html = html.read()
    html = html.decode()
    print(html)
except Exception as err:
    print(err)
```

1.4.2 服务器获取 POST 的数据

服务器用 Flask 中的 `request` 对象的 `form` 来存储 GET 的参数，用 `get` 方法来获取参数，即用 `flask.request.form.get(参数)` 来获取参数的值，例如：

```
province=flask.request.form.get("province")
city=flask.request.form.get("city")
就可以获取 GET 传递的参数 province 与 city 的值。
```

编写服务器程序 `server.py` 如下：

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

@app.route("/",methods=["POST"])
```

```
def index():
    try:
        province=flask.request.form.get("province") if "province" in flask.request.form else
"""
        city = flask.request.form.get("city") if "city" in flask.request.form else ""
        return province+","+city
    except Exception as err:
        return str(err)
```

```
if __name__=="__main__":
```

```
    app.run()
```

值得注意的是在服务器中要指定:

```
@app.route("/",methods=["POST"])
```

```
def index():
```

表明函数这个函数接受 POST 请求。默认时只接受 GET 请求, 如果要接受 POST 请求就必须明确指明, 如果写成:

```
@app.route("/",methods=["GET","POST"])
```

```
def index():
```

那么这个函数即可以接受 GET 请求也可以接受 POST 请求。

我们先运行 server.py 建立 web 网站, 再运行 client.py, 可以看到 client.py 的结果是:
广东,深圳

1.4.3 GET 与 POST 的混合使用

实际上在应用中客户端客户同时使用 GET 与 POST 向服务器发送数据, 一般 GET 的数据时放在地址栏后面的, 参数简单, 数据量少, 而 POST 的数据时表单数据, 数据量大。

例: 客户端向服务器发送一个城市的简介, 服务器接收后返回收到的信息。

我们把省份 province 与城市 city 的名称放在 url 地址栏后面采用 GET 方式发送给服务器, 然后把城市的简介用 POST 方法发送给服务器, 客户端程序如下:

```
import urllib.parse
```

```
import urllib.request
```

```
url="http://127.0.0.1:5000"
```

note="深圳依山傍海, 气候宜人, 实在是适合人类居住的绝佳地。这里四季如春, 干净整洁, 比邻香港, 拥有着丰富的自然景观和人文气息。匆匆过客注意到的也许只有它的时尚繁华, 忙碌的暂居者也可能对它有着不识城市真面目之感。只有世代在此生活的老深圳人, 才默默的看着它从贫穷走向富饶经历了怎样的艰辛。"

```
try:
```

```
    province= urllib.parse.quote("广东")
```

```
    city= urllib.parse.quote("深圳")
```

```
    note= "note="+urllib.parse.quote(note)
```

```
    param="province="+province+"&city="+city
```

```
    html=urllib.request.urlopen("http://127.0.0.1:5000?" +param,data=note.encode())
```

```
    html = html.read()
```

```
    html = html.decode()
```

```
    print(html)
except Exception as err:
    print(err)
```

服务器程序如下:

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

@app.route("/",methods=["GET","POST"])
def index():
    try:
        province=flask.request.args.get("province") if "province" in flask.request.args else ""
        city = flask.request.args.get("city") if "city" in flask.request.args else ""
        note = flask.request.form.get("note") if "note" in flask.request.form else ""
        return province+","+city+"\n"+note
    except Exception as err:
        return str(err)

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

我们先运行服务器程序建立 web 网站,再运行客户端程序,可以看到服务器返回的结果:

广东,深圳

深圳依山傍海,气候宜人,实在是适合人类居住的绝佳地。这里四季如春,干净整洁,比邻香港,拥有着丰富的自然景观和人文气息。匆匆过客注意到的也许只有它的时尚繁华,忙碌的暂居者也可能对它有着不识城市真面目之感。只有世代在此生活的老深圳人,才默默的看着它从贫穷走向富饶经历了怎样的艰辛。

我们看到服务器接收 GET 的参数采用 flask.request.args.get(参数)与接收 POST 参数的方法 flask.request.form.get(参数)是不同的,实际上可以把它们进行统一成 flask.request.values.get(参数),采用这个方法可以获取 GET 的参数也可以获取 POST 的参数,即服务器程序可以进一步改进如下:

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

@app.route("/",methods=["GET","POST"])
def index():
    try:
        province=flask.request.values.get("province") if "province" in flask.request.values
    else ""
        city = flask.request.values.get("city") if "city" in flask.request.values else ""
        note = flask.request.values.get("note") if "note" in flask.request.values else ""
```

```
        return province+","+city+"\n"+note
    except Exception as err:
        return str(err)
```

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

我们采用 `flask.request.values.get(参数)` 后就不再区分参数是 GET 的还是 POST 的了。