
1.3 GET 方法访问网站

访问网站最常用的一种方法是 GET 方法，这种方法主要是客户端从服务器端获取网站数据，如果有交互的话，客户端把参数附加在网址的后面向服务器提供参数，服务器接收这些参数后作出不同的响应。

1.3.1 客户端 GET 方式发送数据

GET 方式发送使得数据附加在 URL 后面，在 URL 后面先接一个“?”号，数据采用：“名称 1=值 1&名称 2=值 2&名称 3=值 3……”

的方式，多个数据之间用“&”符号隔开，例如向服务器传递省份与城市的数据就可以这样写：

```
urllib.request.urlopen("http://127.0.0.1:5000?province=GD&city=SZ")
```

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")
```

```
urllib.request.urlopen("http://127.0.0.1:5000?province="+province+"&city="+city)
```

编写客户端程序 `client.py` 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
```

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

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

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

```
    print(html)
```

```
except Exception as err:
```

```
    print(err)
```

1.3.2 服务器获取 GET 发送的数据

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

```
province=flask.request.args.get("province")
```

```
city=flask.request.args.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("/")
```

```
def index():
```

```
try:
    province=flask.request.args.get("province")
    city = flask.request.args.get("city")
    return province+","+city
except Exception as err:
    return str(err)
```

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

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

注意 web 网址是 <http://127.0.0.1:5000>，如果我们直接访问这个网站会有错误，因为没有提供 province 与 city 参数，服务器执行：

```
province=flask.request.args.get("province")
city = flask.request.args.get("city")
```

语句时会到 args 中查找 province,city 参数的值，结果没有找到而出现错误。为了避免这种错误可以把这两语句写成如下：

```
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 ""
```

这样在"province"、"city"没有出现在 flask.request.args 中时 province、city 值就设置为空字符串。服务器程序改进如下：

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

@app.route("/")
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 ""
        return province+","+city
    except Exception as err:
        return str(err)

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