

# 使用 httplib 抓取

httplib.HTTPConnection ( host [, port [, strict [, timeout ]]])

host 表示服务器主机 port 为端口号, 默认值为 80 strict 的默认值为 false, 表示在无法解析服务器返回的状态行时(status line) (比较典型的状态行如: HTTP/1.0 200 OK), 是否抛 BadStatusLine 异常 可选参数 timeout 表示超时时间。

HTTPConnection 提供的方法:

- HTTPConnection.request ( method, url [, body [, headers ]])

调用 request 方法会向服务器发送一次请求 method 表示请求的方法, 常用有方法有 get 和 post ;

url 表示请求的资源 url ;

body 表示提交到服务器的数据, 必须是字符串 (如果 method 是 " post", 则可以把 body 理解为 html 表 单中的数据) ;

headers 表示请求的 http 头。

- HTTPConnection.getresponse ()

获取 Http 响应。返回的对象是 HTTPResponse 的实例。

- HTTPConnection.connect ()

连接到 Http 服务器。

- HTTPConnection.close () 关闭与服务器的连接。

- HTTPConnection.set\_debuglevel ( level )

设置高度的级别。参数 level 的默认值为 0, 表示不输出任何调试信息。

httplib.HTTPResponse

-HTTPResponse 表示服务器对客户端请求的响应。往往通过调用 HTTPConnection.getresponse()来创建, 它有如下方法和属性:

-HTTPResponse.read([amt]) 获取响应的消息体。如果请求的是一个普通的网页, 那么该方法返回的是页面的 html。可选参数 amt 表示从 响应流中读取指定字节的数据。

-HTTPResponse.getheader(name[, default])

获取响应头。Name 表示头域(header field)名, 可选参数 default 在头域名不存在的情况下作为默认值返回。

-HTTPResponse.getheaders()

以列表的形式返回所有的头信息

-HTTPResponse.msg

获取所有的响应头信息。

-HTTPResponse.version

获取服务器所使用的 http 协议版本。11 表示 http/1.1; 10 表示 http/1.0。

-HTTPResponse.status

获取响应的状态码。如: 200 表示请求成功。

-HTTPResponse.reason

返回服务器处理请求的结果说明。一般为 " OK"

```
#!/usr/bin/python
```

```
# -*- coding:utf-8 -*-
```

```
def use_httplib():
```

```
import httplib
```

```
conn = httplib.HTTPConnection("www.baidu.com")
```

```
    i_headers = {"User-Agent": "Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN; rv:1.9.1) Gecko/20090624  
Firefox/3.5", "Accept": "text/plain"}
```

```
    conn.request("GET", "/", headers=i_headers)
```

```
    r1 = conn.getresponse()
```

```
    print "version:", r1.version
```

```
    print "-----"
```

```
    print "reason:", r1.reason
```

```
    print "-----"
```

```
    print "status:", r1.status
```

```
    print "-----"
```

```
    print "msg:", r1.msg
```

```
    print "-----"
```

```
    print "headers:", r1.getheaders()
```

```
    print "-----"
```

```
    data = r1.read()
```

```
    print data.decode("utf-8")
```

```
print "-----"  
print "data length:%s"%len(data)  
conn.close()
```