### 概述

http.client模块定义和实现了一系列类用于实现客户端HTTP和HTTPS协议。

一句话说明: HTTP/1.1 client library

注:

- 1. 支持HTTP/1.1版本的协议
- 2. HTTPS的支持需要安装SSL才行

## http.client工作流程

下面我们看一下http.client工作流程机制,以便加深在原理的理解。

```
(null)
   | HTTPConnection() 建立HTTP链接
Idle
   | putrequest() 准备请求内容
Request-started
   | ( putheader() )* endheaders() 准备请求头
Request-sent 发送请求
   | getresponse() raises 获取服务端响应
| response = getresponse() | ConnectionError
Unread-response Idle
[Response-headers-read] 响应头读取
   response.read()
                     | putrequest() 自己往下看吧
Idle
               Req-started-unread-response
                    response.read()
   Request-started
                   Req-sent-unread-response
                     response.read()
                    Request-sent
```

#### 说明:

- 1. HTTPConnection通过不同的"状态"定义了HTTP客户的行为状态,管理维护着每一个的request和response
- 2. 上述的流程图,详细的说明了各"状态"之间的转换,把上图理解,即把http.client理解透彻了。

### http.client的类

下面我们看一下在http.client模块中,定义了哪些类

- HTTPConnection
  - 一个HTTPConnection实例代表一个与HTTP服务器的事务。
- HTTPSConnection HTTPConnection的子类,它使用SSL与安全的服务器进行通信。
- HTTPResponse HTTPResponse的实例代表客户端与服务端成功建立链接后的返回。
- HTTPException httpc.client模块中的异常基类,其为Exception的子类。

以下均为HTTPException的子类,属于异常类,具体含义这里不一一说明了,大家根据字面意思了解即可

- NotConnected
- InvalidURL
- UnknownProtocol
- UnknownTransferEncoding
- UnimplementedFileMode
- IncompleteRead
- ImproperConnectionState
- CannotSendRequest
- CannotSendHeader
- ResponseNotReady
- BadStatusLine
- LineTooLong
- RemoteDisconnected

#### 定义的几个常量:

- http.client.HTTP\_PORT HTTP默认端口(80)。
- http.client.HTTPS\_PORT https默认端口(443)。
- http.client.responses HTTP 1.1状态码映射字典。

例如:

http.client.responses[http.client.NOT\_FOUND] is 'Not Found'

注: 更多的状态码请参加https://docs.python.org/3/library/http.html#http-status-codes

# 小结

2/3 公众号: 开源优测 - Focus on open source solutions in the field of software testing

扫一扫关注微信公众号:

