21.24. http.cookiejar –处理HTTP客户端的Cookie

源代码: Lib/httlp/cookiejar.py

http.cookiejar模块定义了自动处理HTTP cookie的类。它是用来进入一些需要cookie（通过来自服务器的HTTP response设置在客户机上）的网站.

Netscape cookie协议和定义在RFC 2965上的协议都可以被处理。RFC 2965的处理默认关闭。RFC 2965 cookie像Netscape cookie一样被分析，随后根据实际“政策”来当作RFC 2965或者Netscape来处理。

http.cookiejar尝试遵循实际的Netscape cookie协议（与当初设立时的Netscape协议有所不同），包括从RFC 2965引进的maxage和port的cookie属性。

**注意：**在Set-Cookie和Set-Cookie2的请求头中命名的参数（比如domain和expires）通常被当做属性。为了让它们与Python的属性区别开来，这个模块的文档用cookie-attribute来代替。

这个模块定义了如下异常：

**exception http.cookiejar.LoadError**

当从文件中载入cookie失败时由FileCookieJar实例抛出的异常。LoadError是OSError的子类。

3.3版本变动：LoadError不是IOError的子类了。

提供了下面的类：

**class http.cookiejar.CookieJar(policy=None)**

policy是一个CookiePolicy实现的接口对象

CookieJar类存储HTTP cookies。它从HTTP request中提取出cookie，然后在HTTP responses中返回它们。CookieJar实例在必要的时候会自动销毁cookies。它的子类也能从文件或者数据库中存储或者读取cookies。

**class http.cookiejar.FileCookieJar(finename,delayload=None, policy=None)**

policy是一个CookiePolicy实现的接口对象。对于其它参数，看文档中相应的属性。

CookieJar可以从文件或磁盘中存储或读取cookie。除非load()或者server()方法被调用，否则cookies不会从文件中载入。这个类的子类在FileCookieJar subclass and co-operation with web browsers的部分中给出介绍。

**class http.cookiejar.CookiePolicy**

这个类是用来决定cookie是否要从服务器接收/返回cookie

**class http.cookiejar.DefaultCookiePolicy(blocked\_domains=None, allowed\_domains=None,netscape=True,rfc2965=False,rfc2109\_as\_netscape=None,hide\_cookie2=False,stric\_domain=False,strict\_rfc2965\_unverifiable=True,strict\_ns\_unverifiable=False,strict\_ns\_domain=DefaultCookiePolicy.DomainLiberal,strict\_ns\_set\_initial\_dollar=False,strict\_ns\_set\_path=False)**

构造参数只能被当做关键词参数传递。blocked\_domains是一个我们不想接收cookie的域名序列，当然也不会向那些域名返回cookie。allowed\_domains如果不为空，那么这里面的序列是我们只能从给定的域名接收和返回的cookie域名列表。对于其它参数，参阅CookiePolicy和DefaultCookiePolicy的文档。

DefaultCookiePolicy实现了Netscape和RFC 2965的接收/拒绝规则。RFC 2109cookie默认以RFC 2965的规则对待（比如：从Set-Cookie头中接收的cookie-attribute版本号为1）。但是，如果RFC 2965处理被关闭或者rfc2109\_as\_netscape为True，RFC2109的cookie会被CookieJar的实例降级成Netscape cookie通过设置Cookie实例的版本属性为0。DefaultCookiePolicy也提供一些允许调整规则的参数。

21.24.2 FileCookieJar的子类和浏览器的协同

以下的CookieJar子类是用来读取和存储的

*Class http.cookiejar.MozillaCookieJar(filename,delayload=None,policy=None)*

FileCookieJar能从磁盘的Mozilla里的cookies.txt（被Lynx和Netscape使用）读取和存储cookies。

注意：这将会损失RFC 2965 cookies，新的非标准的cookie属性的信息。

*Class http.cookiejar.LWPCookieJar(filename,dalayload=None,policy=None)*

一个能从磁盘中读取或存储cookies并且与libwww-perl库的Set-Cookies相兼容的文件格式的FileCookieJar。如果你想存储易于人阅读的cookies这个子类将会很方便。

21.6 urllib.request --打开URL的第三方库

提供了如下函数：

*urllib.request.urlopen(url,data=None,[timeout]\*,cafile=None,capath=False,context=None)*

打开url连接，url可以为string或者一个Request对象

*data*必须是字节对象，用来说明向服务器发送额外的数据，如果不需要发送数据的话就为None。data也可以为一个可迭代对象，这样的话Content\_Length的值必须要在headers说明。目前的HTTP请求是唯一需要用data的。如果data参数给出的话HTTP请求就为POST而不是GET。

urllib.request模块使用HTTP/1.1并且在HTTP请求中包含Connection:close的请求头。

可选的timeout参数说明操作阻塞（比如尝试连接）的话超时多少秒。它只在HTTP,HTTPS,FTP链接中工作。

如果context给出了，它必须是一个ssl.SSLContext的实例用来说明不同的SSL选项。参阅 HTTPSConnection

\*可选的cafile和capath参数是关于证书的。

\*

对于http和https的url，这个函数返回一个http.client.HTTPResponse对象，它有HTTPResponse Objects的方法。

\*对于ftp,

提供了下面的类：

class *urllib.request.Request(url,data=None,headers={},origin\_req\_host=None,unverifiable=False,method=None)*

这个类是一个URL请求的抽象。

url应该是一个包含合法URL的string

*data*必须是字节对象，用来说明向服务器发送额外的数据，如果不需要发送数据的话就为None。目前的HTTP请求是唯一需要用data的。如果data参数给出的话HTTP请求就为POST而不是GET。

Urllib.parse.urlencode()函数接收一个映射类型或者二元序列，返回一个字符串。在data参数被使用前它必须被编码成字节。

headers应该为一个字典。

*urllib.request.install\_opener(opener)*

安装一个OpenerDirector实例来作为默认全局的opener。如果你想使用opener来打开url的话才有必要安装，简单的调用OpenerDirector.open()而不是urlopen()。代码不会对一个真的OpenerDirector检查，任何有合适的接口的类都可以工作。

*urllib.request.build\_opener([handler,...])*

返回一个顺序给出的handler的OpenerDirector实例.handlers可以是BaseHandler的实例，或者是BaseHandler的子类（必须不需要任何参数来调用构造函数）。下面类的实例应该在handlers的前面，除非handlers包含它们：ProxyHandler(如果代理设置被检测到的话),UnknownHandler,HTTPHandler,HTTPDefaultErrorHandler,HTTPRedirecHandler,FTPHandler,FileHandler,HEEPErrorProcessor。

如果Python支持SSL supprt(比如，能import ssl),HTTPSHandler也应该被添加。

Base\_Handler的子类也可以改变它的handler\_order的属性来改变它在handlers表中的位置

*class urllib.requset.OpenerDirector*

OpenerDirector类通过BaseHandleer来打开URL。它管理hanlers链条，并且从错误中恢复。

*class urllib.request.HTTPCookieProcessor(cookiejar=None)*

用来处理HTTP Cookies的类