

urlopen(url, data=None, timeout=socket.GLOBAL_DEFAULT_TIMEOUT, *, cafile=None, capath=None, cadefault=False, context=None)

url: 请求的url。

data: 请求的data, 如果设置了这个值, 那么将变成post请求。

cafile、capath、cadefault 参数: 用于实现可信任的CA证书的HTTP请求。(基本上很少用)

context参数: 实现SSL加密传输。(基本上很少用)

返回值: 返回值是一个http.client.HTTPResponse对象, 这个对象是一个类文件句柄对象。有read(size)、readline、readlines以及getcode等方法。

read(size): 返回指定字节的数目, 默认为全部 (返回的是二进制文件哦) resp.read().decode()可以进行解码

readline(): 返回一行

readlines(): 用列表返回全部行

getcode(): 返回状态码, resp.getcode()等价于resp.status

getheaders(): 返回响应头 Return list of (header, value) tuples.

urlretrieve(url, filename=None, reporthook=None, data=None)

参数url: 下载链接地址

参数filename: 指定了保存本地路径 (如果参数未指定, urllib会生成一个临时文件保存数据。)

参数reporthook: 是一个回调函数, 当连接上服务器、以及相应的数据块传输完毕时会触发该回调, 我们可以利用这个回调函数来显示当前的下载进度。

参数data: 指post到服务器的数据, 该方法返回一个包含两个元素的(filename, headers) 元组, filename 表示保存到本地的路径, header表示服务器的响应头

用来下载网页, 或者下载图片。

URLError产生的原因

- 1). 网络无连接
- 2). 连接不到特定的服务器
- 3). 服务器不存在

URLError

URLError含有属性: reason

HTTPError

HTTPError是URLError的子类。

HTTPError含有属性: code, reason, headers

parse.urlencode函数

urlencode可以把字典数据转换为URL编码的数据

parse.urlencode({'wd': '刘德华'})
==> wd=%E5%88%98%E5%BE%B7%E5%8D%8E

parse.parse_qs函数

将经过编码后的url参数进行解码

parse.parse_qs('wd=%E5%88%98%E5%BE%B7%E5%8D%8E')
==> {'wd': ['刘德华']}

parse.urlparse和parse.urlsplit函数

对url中的各个组成部分进行分割

urlparse和urlsplit基本上是一模一样的。

唯一不一样的地方是, urlparse里面多了一个params属性

parse.urlunparse函数

对url中的各个组成部分进行组合

parse.urljoin函数

用来拼接两个网页

增加请求头

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT; x64) '}
req = request.Request(url, headers=headers)
request.urlopen(req)

(4) request.Request类

Post请求

data = {'user': 'root', 'password': '123456'} (数据先存入字典)
data = parse.urlencode(data).encode('utf-8') (传入的data一定要是bytes类型)
req = request.Request(url, data = data, headers = headers, method='POST')
resp = request.urlopen(req, timeout = 5)

(5) ProxyHandler处理器 (代理设置)

handler = urllib.request.ProxyHandler({'https': '136.228.129.36:41838'})
opener = urllib.request.build_opener(handler)
resp = opener.open(url)

(6) cookie的处理

request.HTTPCookieProcessor

http.cookiejar

urllib库

(1) urllib.request库

request.urlopen函数

request.urlretrieve函数

(2) 异常处理urllib.error

(3) urllib.parse库

第一处:

```
from urllib import request, error
import socket
try:
    response = request.urlopen('http://aaa.com', timeout=3)
except error.HTTPError as e:
    print("The server couldn't fulfill the request.")
    print('Error code: ', e.code)
except error.URLError as e:
    print('We failed to reach a server.')
    print('Reason: ', e.reason)
    if isinstance(e.reason, socket.timeout):
        print('连接超时')
else:
    print('Request Successfully')
```

第二处:

```
from urllib import request, parse
#resp = request.urlopen('http://www.baidu.com/s?wd=刘德华')
#UnicodeEncodeError: 'ascii' codec can't encode characters in
position 10-12: ordinal not in range(128)
parms = {'wd': '刘德华'}
url = 'http://www.baidu.com/s'
query_string = parse.urlencode(parms)
print(query_string)
url = url+'?' + query_string
resp = request.urlopen(url)
print(resp.getcode())
```

第三处:

```
from urllib import request, parse
from http.cookiejar import CookieJar
headers = {'User-Agent': "Mozilla/5.0 (Macintosh; U; Mac OS X Mach-O;
en-US; rv:2.0a) Gecko/20040614 Firefox/3.0.0 "}
def get_opener():
    cookjar = CookieJar()
    handler = request.HTTPCookieProcessor(cookjar)
    opener = request.build_opener(handler)
    return opener
def save_cookie(opener):
    data = {"email": "970138074@qq.com", "password": "pythonspider"}
    login_url = "http://www.renren.com/PLogin.do"
    req = request.Request(login_url, data=
parse.urlencode(data).encode('utf-8'), method='POST', headers=headers)
    opener.open(req)
def get_profile(opener):
    profile_url = 'http://www.renren.com/880151247/profile'
    req = request.Request(profile_url, headers=headers)
    resp = opener.open(req)
    print(resp.read().decode())
if __name__ == '__main__':
    opener = get_opener()
    save_cookie(opener)
    get_profile(opener)
```