

目录

前言	1.1
Python网络库概览	1.2
内置网络库	1.3
urllib	1.3.1
第三方网络库	1.4
requests	1.4.1
使用心得	1.4.1.1
通用	1.5
常用的User-Agent	1.5.1
附录	1.6
参考资料	1.6.1

Python心得：http网络库

- 最新版本： `v1.0`
- 更新时间： `20210703`

简介

总结Python中的网络库，包括内置的urllib，以及第三方库requests，和通用的内容，比如常见User-Agent用户代理等。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

Gitbook源码

- [crifan/python_summary_http_lib: Python心得：http网络库](#)

如何使用此Gitbook源码去生成发布为电子书

详见：[crifan/gitbook_template: demo how to use crifan gitbook template and demo](#)

在线浏览

- Python心得：[http网络库 book.crifan.com](#)
- Python心得：[http网络库 crifan.github.io](#)

离线下载阅读

- Python心得：[http网络库 PDF](#)
- Python心得：[http网络库 ePub](#)
- Python心得：[http网络库 Mobi](#)

版权说明

此电子书教程的全部内容，如无特别说明，均为本人原创和整理。其中部分内容参考自网络，均已备注了出处。如有发现侵犯您的版权，请通过邮箱联系我 `admin` 艾特 `crifan.com`，我会尽快删除。谢谢合作。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 crifan 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

更多其他电子书

本人 crifan 还写了其他 100+ 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme: Crifan的电子书的使用说明](#)

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新： 2021-07-03 10:44:38

Python网络库概览

之前已有[提到](#)，关于网络Http，有很多相关的工具，以及不同开发语言各自有很多库。

此处就来整理，Python语言的[网络相关的库](#)。

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved,
powered by Gitbook最后更新：2021-07-03 09:50:20

内置网络库

关于网络方面，Python有个内置的，功能很强大的库：[urllib](#)

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved,
powered by Gitbook最后更新：2021-07-03 09:53:00

urllib

此处先介绍Python的内置网络库的发展历史：

- 背景：Python有2个大的版本： Python2 和 Python3
- Python2
 - 有独立的2个库： [urllib](#) 和 [urllib2](#)
 - 缺点：功能划分的不是很清楚
- Python3
 - 所以合并了 [urllib](#) 和 [urllib2](#) 为单个库： [urllib](#)
 - 其中： Python2 的 [urllib2](#) 合并成 Python3 的 [urllib.request](#)

crifan的Python的网络函数

自己很早之前在折腾爬虫期间，就利用(Python2的) [urllib](#) 总结出了自己的获取 http 的 [response](#) 或 [html](#) 的函数：

- [Python2 crifanLib.py](#)
 - [getUrlResponse](#)
 - [getUrlRespHtml](#)

且还给出了详细的[说明文档](#)

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved,
powered by Gitbook最后更新： 2021-07-03 10:06:33

第三方网络库

Python中有很多第三方的网络库。其中最出名的最流行的要属：`requests`

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新: 2021-07-03 10:07:28

requests

- requests
 - 概述：为人类用起来很方便而开发的Python的网络库。是目前最流行的Python的网络库
 - GitHub
 - psf/requests: A simple, yet elegant HTTP library
 - <https://github.com/psf/requests>
 - 官网文档
 - <https://docs.python-requests.org>

如何使用requests

官网文档，已经写的足够好了，就不重复了。详见：

- requests官网资料
 - 英文
 - Quickstart — Requests 2.11.1 documentation
 - <http://docs.python-requests.org/en/master/user/quickstart/#make-a-request>
 - 中文
 - Requests: 让 HTTP 服务人类 — Requests 2.18.1 文档
 - http://cn.python-requests.org/zh_CN/latest/
 - 快速上手 — Requests 2.18.1 文档
 - http://cn.python-requests.org/zh_CN/latest/user/quickstart.html
 - host在readthedocs
 - Quickstart — Requests 2.23.0 documentation
 - <https://requests.readthedocs.io/en/master/user/quickstart/>

此处仅仅（摘录）列举最简单的内容，供了解：

安装

```
pip install requests
```

基本用法举例

导入，get请求：


```
import requests

r = requests.get('https://api.github.com/events')
```

POST带参数，内部会自动把 dict 转换成 json：

```
r = requests.post('https://httpbin.org/post', data = {'key
```

GET带参数，内部自动转换 dict 为 key=value 的 query parameter

```
payload = {'key1': 'value1', 'key2': 'value2'}
r = requests.get('https://httpbin.org/get', params=payload)
```

返回的response:

```
r = requests.get('https://api.github.com/events')
r.text
# '[{"repository":{"open_issues":0,"url":"https://github.c
```

response有各种属性可用:

查看字符编码:

```
r.encoding
# 'utf-8'
```

手动设置编码:

```
r.encoding = 'ISO-8859-1'
```

返回内容:

```
r.content
# b' [{"repository":{"open_issues":0,"url":"https://github.c
```

等等。

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新: 2021-07-03 10:43:20

requests使用心得

自己在使用requests期间，有些心得和总结，整理如下：

crifan的requests总结

把基于requests的常用功能，封装成函数，供需要的参考：

<https://github.com/crifan/crifanLibPython/blob/master/python3/crifanLib/t hirdParty/crifanRequests.py>

其他心得

get加上参数

举例：

```
payload = {'key1': 'value1', 'key2': 'value2'}
r = requests.get('https://httpbin.org/get', params=payload)
print(r.url)
# https://httpbin.org/get?key2=value2&key1=value1
```

header和post的body

举例：

```
gHeaders = {
    "Authorization": "JWT xxx", # JWT eyJ0xxxxxxxixJ9.xxxxxxx
}

postBody = {
    "username": Username,
    "password": Password,
}

getTokenResp = requests.post(GetJwtTokenUrl, data=postBody)

saveScriptResp = requests.post(CreateScriptUrl, headers=gHeaders)
```

传入post的data时，不要dict的json，而是要str的json字符串

注：对标常见情况是，自己把dict的用dict转换成json字符串，再给data

```
import requests
import json

scripJsonStr = json.dumps(curScriptDict)
saveScriptResp = requests.post(CreateScriptUrl, headers=gHe
```

或:

或者把dict传递给json, requests自动帮你转换成json

```
import requests

saveScriptResp = requests.post(CreateScriptUrl, headers=gHe
```

-> 说明不是两者同时传递的参数, 而是二选一

传递json类型参数时, 记得加上content-type的header

```
gHeaders = {
    'Content-Type': 'application/json; charset=utf-8',
    "Accept": 'application/json',
}
```

否则可能会导致 [TypeError string indices must be integers](#)

get返回二进制, 保存到文件

下载文件 下载二进制文件

```
import requests
resp = requests.get(pictureUrl)
with open(saveFullPath, 'wb') as saveFp:
    saveFp.write(resp.content)
```

举例:

```
gHeaders = {
    "User-Agent": UserAgent_Mac_Chrome,
}
curPictureUrl = "https://bp.pep.com.cn/ebook/yybanjxc/file:
saveFullPath = "output/义教教科书英语八年级下册/mobile_1.jpg"
resp = requests.get(curPictureUrl, headers=gHeaders)
if resp.ok:
    with open(saveFullPath, 'wb') as saveFp:
        saveFp.write(resp.content)
```

上传multipart/form-data表单参数和文件

上传multipart/form-data表单form参数

curl 命令:

```
curl -X PUT http://127.0.0.1:8080/api/xxx ...
-H 'content-type: multipart/form-data; boundary=----xxx' \
-F taskStatus=1
```

对应 Python 的 requests 的: [More complicated POST requests](#)

代码:

```
updateTaskUrl = "http://127.0.0.1:8080/api/xxx"
updateInfoDict = {
    "taskStatus": 1,
}
resp = requests.put(updateTaskUrl, data=updateInfoDict)
```

上传multipart/form-data文件

curl 命令:

```
curl -X POST http://127.0.0.1:8080/api/xxx ...
-H 'content-type: multipart/form-data; boundary=----xxx' \
-F file=@/Users/xxx.txt
```

对应 Python 的 requests 的: [POST a Multipart-Encoded File](#)

代码:

```
filePath = "/Users/xxx.txt"
fileFp = open(filePath, 'rb')
fileInfoDict = {
    "file": fileFp,
}
resp = requests.post(uploadResultUrl, files=fileInfoDict)
```

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook 最后更新: 2021-07-03 10:23:25

通用

此处整理，Python处理网络相关内容会涉及到的一些通用的知识和心得。

crifan的http相关库函数

自己在使用http相关网络功能期间，总结出一些常用功能，封装成函数。供参考：

<https://github.com/crifan/crifanLibPython/blob/master/python3/crifanLib/crifanHttp.py>

包括有：

- randomUserAgent
- isFileValid
- downloadFile
- manuallyDownloadFile
- getUrlResponse
- getUrlRespHtml
- getUrlRespHtml_multiTry
- 等

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新：2021-07-03 10:32:43

常用的User-Agent

写(Python)代码，处理网络请求期间，常涉及到，模拟某种浏览器，发出请求，以希望返回希望的页面内容。

期间涉及到，传递Header参数 User-Agent = 用户代理

此处整理一些常见的User-Agent值，供参考。

```
constUserAgentList = [  
    ### PC  
    # chrome  
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.87 Safari/537.36"  
    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.87 Safari/537.36"  
    "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/73.0.3683.87 Safari/537.36"  
    "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.87 Safari/537.36"  
    # Firefox  
    "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0"  
    "Mozilla/5.0 (X11; U; Linux x86_64; zh-CN; rv:1.9.2.10) Gecko/20100101 Firefox/3.0.10"  
    # Safari  
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_2) AppleWebKit/537.36 (KHTML, like Gecko) Version/12.0 Safari/537.36"  
    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.57.24 (KHTML, like Gecko) Version/5.0 Safari/534.57.24"  
    # 360  
    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.87 Safari/537.36"  
    "Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko"  
    # QQ浏览器  
    "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; QQBrowser/9.0.0.1024)"  
    "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; QQBrowser/6.0.2691.401; Trident/4.0)"  
    # sogou浏览器  
    "Mozilla/5.0 (Windows NT 5.1) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/27.0.1453.94 Safari/535.11"  
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; SogouBrowser/4.0.0.1024)"  
    # UC浏览器  
    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.87 Safari/537.36"  
]
```

最新代码详见：

<https://github.com/crifan/crifanLibPython/blob/master/python3/crifanLib/crifanHttp.py>

随机获取User-Agent

想要从上述列表中，随机获取一个（以模拟普通人通过浏览器访问网页，防止被反扒），对应代码是：

```
import random

def randomUserAgent():
    """random get a common User-Agent"""
    global constUserAgentList
    curUA = random.choice(constUserAgentList)
    return curUA
```

最新代码详见：

<https://github.com/crifan/crifanLibPython/blob/master/python3/crifanLib/crifanHttp.py>

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新： 2021-07-03 10:34:02

附录

下面列出相关参考资料。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved,
powered by Gitbook最后更新: 2020-03-17 09:11:34

参考资料

- [【已解决】爬取bp.pep.com.cn中的义务教育教科书资源](#)
- [【已解决】Python的requests中如何下载二进制数据保存为图片文件](#)
- [HTTP的工具和库 · HTTP知识总结](#)
- [【已解决】Python中如何模拟用户登录和传递JWT的token以获取访问API的权限](#)
- [【已解决】Python接口返回400错误: {'detail': 'JSON parse error - Expecting value: line 1 column 1 \(char 0\)'}](#)
- [【已解决】Python调用接口出错: TypeError string indices must be integer](#)
- [python2和python3中的urllib - 简书](#)
- [More complicated POST requests](#)
- [POST a Multipart-Encoded File](#)
- [python requests的安装与简单运用 - 张亚楠 ' BLOG - 天意从来高难问](#)
- [Python+Requests编码识别Bug](#)
- [代码分析Python requests库中文编码问题 | 峰云就她了](#)
- [Python HTTP库requests中文页面乱码解决方案! - bitpeng - 博客园](#)
-

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新: 2021-07-03 10:33:43