



网络请求

—— urllib库和requests库的使用

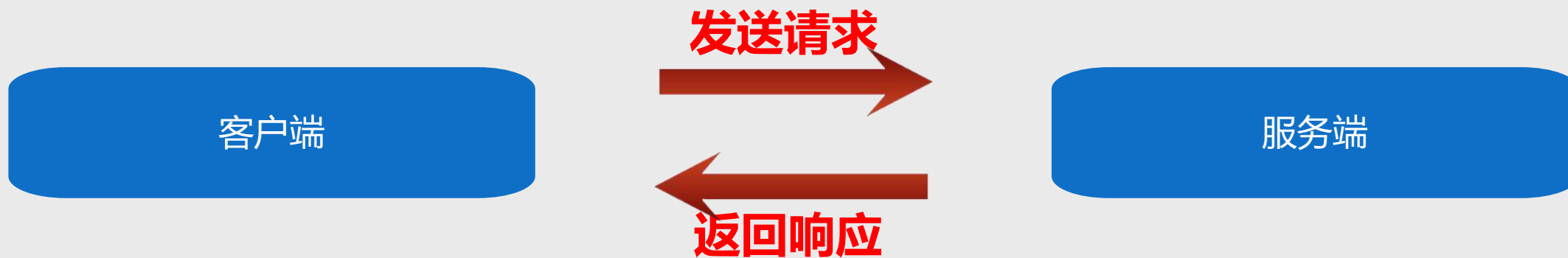
讲师：王老师

1. urllib库
2. requests库

-  1. 了解urllib库
-  2. 熟练掌握urllib函数
-  3. 熟悉urllib.Request，处理cookie，代理设置

urllib库是Python中一个最基本的网络请求库。可以模拟浏览器的行为，向指定的服务器发送一个请求，并可以保存服务器返回的数据。

PS：
urllib是Python自带的标准库，无需安装，直接可以用。



urlopen函数：

在Python3的urllib库中，所有和网络请求相关的方法，都被集到**urllib.request**模块下面了，以先来看下urlopen函数基本的使用：

```
from urllib import request  
resp = request.urlopen('http://www.baidu.com')  
print(resp.read())
```

一个基本的url请求对应的python代码真的非常简单。

urlopen函数详解：

创建一个表示远程url的类文件对象，然后像本地文件一样操作这个类文件对象来获取远程数据。

1. url：请求的url。
2. data：请求的数据，如果设置了这个值，那么将变成post请求。
3. 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。

```
def urlopen(url, data=None, timeout=socket._GLOBAL_DEFAULT_TIMEOUT,  
            *, cafile=None, capath=None, cadefault=False, context=None):  
    '''Open the URL url, which can be either a string or a Request object.  
  
    *data* must be an object specifying additional data to be sent to  
    the server, or None if no such data is needed. See Request for  
    details.
```

urlretrieve函数：

这个函数可以方便的将网页上的一个文件保存到本地。以下代码可以非常方便的将百度的首页下载到本地：

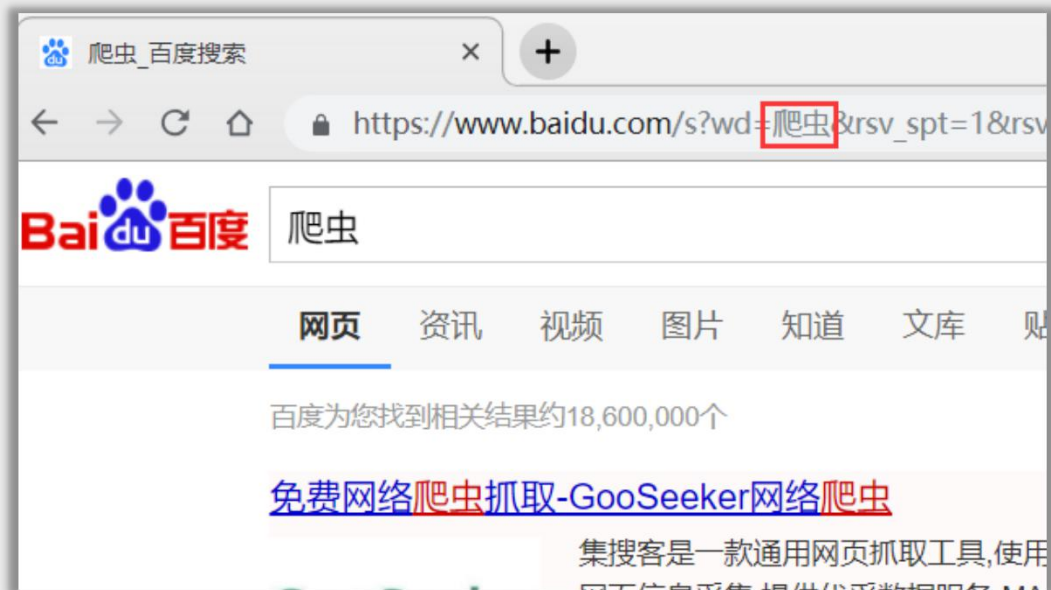
```
from urllib import request  
request.urlretrieve('http://www.baidu.com/', 'baidu.html')
```

urlencode函数：

urlencode可以把字典数据转换为URL编码的数据。

示例代码如下：

```
from urllib import parse
data = {'name':'爬虫基础','greet':'hello world','age':100}
qs = parse.urlencode(data)
print(qs)
```



parse_qs函数：

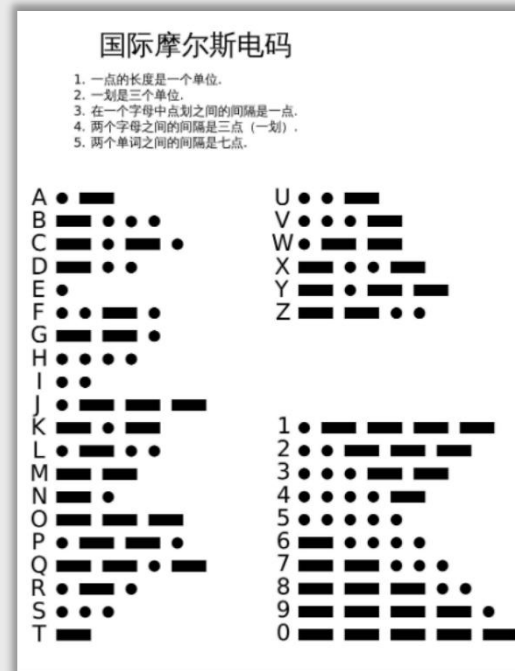
可以将经过编码后的url参数进行解码。

示例代码如下：

```
from urllib import parse
```

```
qs="name=%E7%88%AC%E8%99%AB%E5%9F%BA%E7%A1%80&greet=hello+wo  
rld&age=100"
```

```
print(parse.parse_qs(qs))
```



urlparse和urlsplit :

有时候拿到一个url，想要对这个url中的各个组成部分进行分割，那么这时候就可以使用urlparse或者是urlsplit来进行分割。



urlparse和urlsplit基本上是一模一样的。

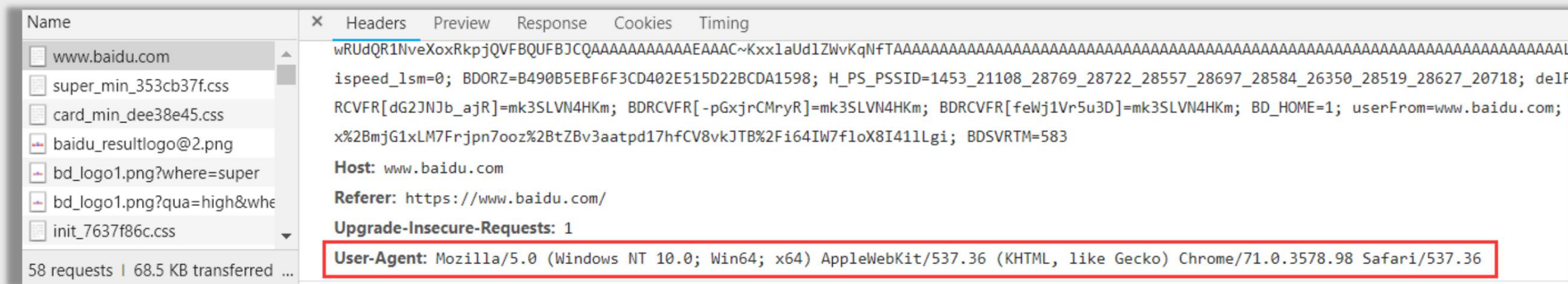
唯一不一样的地方是：

urlparse里有params属性，而urlsplit没有这个params属性。

request.Request类：

如果想要在请求的时候增加一些请求头，那么就必须使用request.Request类来实现。

比如要增加一个User-Agent



演示内容介绍

爬取猫眼实时票房

注意事项

- 请求头
- 响应获取
- URL





- 必做内容

爬取别逗了笑话网

- 选做内容

爬出多页

- 网址

<https://www.biedoul.com/>

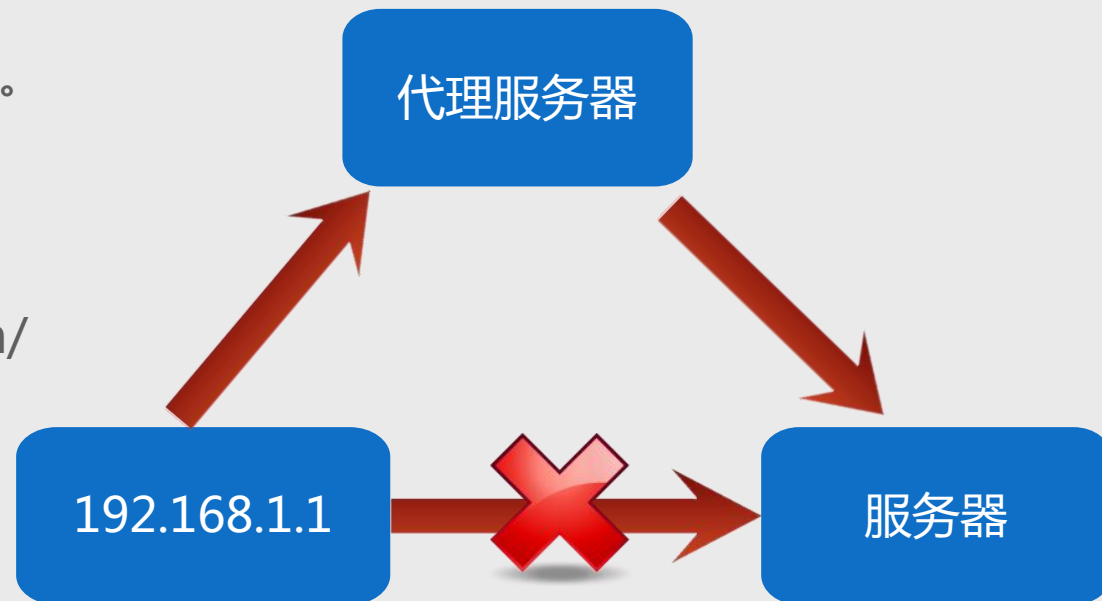
ProxyHandler处理器（代理设置）

很多网站会检测某一段时间某个IP的访问次数(通过流量统计，系统日志等)，如果访问次数多的不像正常人，它会禁止这个IP的访问。

<http://httpbin.org>：查看http请求的一些参数。

常用的代理有：

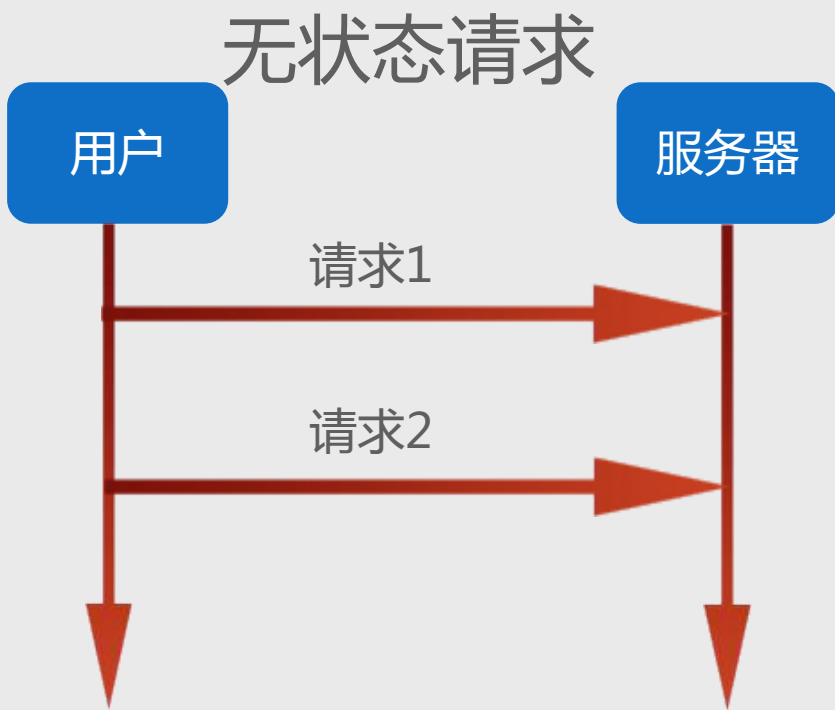
- 西刺免费代理IP：<http://www.xicidaili.com/>
- 快代理：<http://www.kuaidaili.com/>
- 代理云：<http://www.dailiyun.com/>



什么是cookie：

指某些网站为了辨别用户身份、进行 session 跟踪而储存在用户本地终端上的数据

cookie存储的数据量有限，不同的浏览器有不同的存储大小，但一般不超过4KB。因此使用cookie只能存储一些小量的数据。



cookie的格式：

Set-Cookie: NAME=VALUE ; Expires/Max-age=DATE ; Path=PATH ;
Domain=DOMAIN_NAME ; SECURE

参数意义：

NAME：cookie的名字。

VALUE：cookie的值。

Expires：cookie的过期时间。

Path：cookie作用的路径。

Domain：cookie作用的域名。

SECURE：是否只在https协议下起作用。

演示内容介绍

爬虫使用cookie模拟登陆

注意事项

- http.cookiejar模块
- 保存cookie到本地
- 加载cookie



http.cookiejar模块：

该模块主要的类有CookieJar、FileCookieJar、MozillaCookieJar、LWPCookieJar。
这四个类的作用分别如下：

1. CookieJar：管理HTTP cookie值、存储HTTP请求生成的cookie、向传出的HTTP请求添加cookie的对象。整个cookie都存储在内存中，对CookieJar实例进行垃圾回收后cookie也将丢失。
2. FileCookieJar (filename,delayload=None,policy=None)：从CookieJar派生而来，用来创建FileCookieJar实例，检索cookie信息并将cookie存储到文件中。
filename是存储cookie的文件名。delayload为True时支持延迟访问访问文件，即只有在需要时才读取文件或在文件中存储数据。

3. MozillaCookieJar (filename,delayload=None,policy=None) : 从FileCookieJar派生而来, 创建与Mozilla浏览器 cookies.txt兼容的FileCookieJar实例。
4. LWPCookieJar (filename,delayload=None,policy=None) : 从FileCookieJar派生而来, 创建与libwww-perl标准的 Set-Cookie3 文件格式兼容的FileCookieJar实例。

1. 了解requests库



2. 熟练掌握requests库基本使用



3. 熟悉使用代理，处理cookie等

Requests：让HTTP服务人类

虽然Python的标准库中 urllib模块已经包含了平常我们使用的大多数功能，但是它的API使用起来让人感觉不太好，而 Requests宣传是 “HTTP for Humans”，说明使用更简洁方便。

Requests 是用Python语言编写，基于 urllib，但是它比 urllib 更加方便，可以节约我们大量的工作，完全满足 HTTP 测试需求。

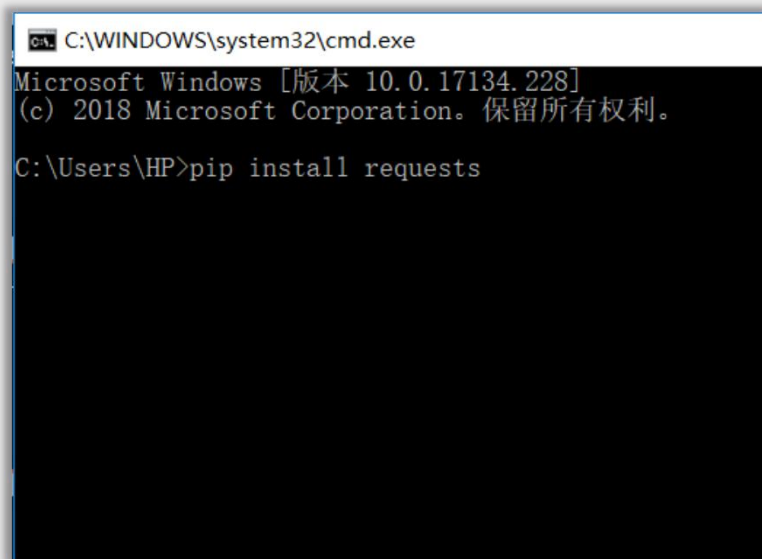
安装和文档地址：

利用pip可以非常方便的安装：

```
pip install requests
```

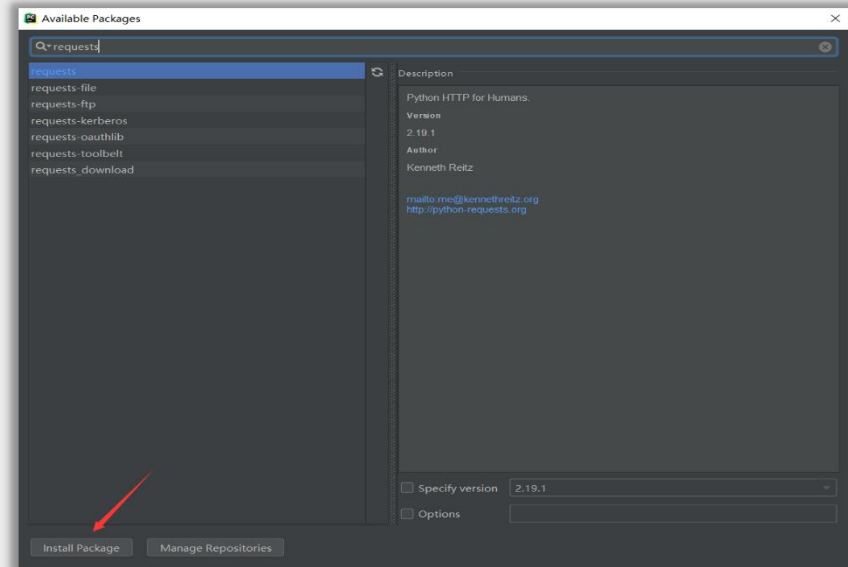
中文文档：http://docs.python-requests.org/zh_CN/latest/index.html

github地址：<https://github.com/requests/requests>



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.17134.228]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\HP>pip install requests
```



发送GET请求：

最简单的发送get请求就是通过requests.get来调用：

```
response = requests.get("http://www.baidu.com/")
```

发送POST请求：

最基本的POST请求可以使用post方法：

```
response = requests.post("http://www.baidu.com/", data=data)
```


使用代理：

使用requests添加代理也非常简单，只要在请求的方法中（比如get或者post）传递proxies参数就可以了。



人在江湖飘、哪能不挨刀。
我是吕子乔、保命用小号。

cookie :

如果在一个响应中包含了cookie，那么可以利用cookies属性拿到这个返回的cookie值

```
import requests
```

```
resp = requests.get('http://www.baidu.com/')
```

```
print(resp.cookies)
```

```
print(resp.cookies.get_dict())
```

session :

使用requests，也要达到共享cookie的目的，那么可以使用requests库给我们提供的session对象。

注意：这里的session不是web开发中的那个session，这个地方只是一个会话的对象而已。

处理不信任的SSL证书：

对于那些已经被信任的SSL证书的网站，比如<https://www.baidu.com/>，那么使用requests直接就可以正常的返回响应。示例代码如下：

```
resp = requests.get('https://inv-veri.chinatax.gov.cn/', verify=False)
print(resp.content.decode('utf-8'))
```



- 必做内容

使用requests库爬取糗事百科

- 选做内容

爬取前10页

EDU

CSDN学院 IT实战派

