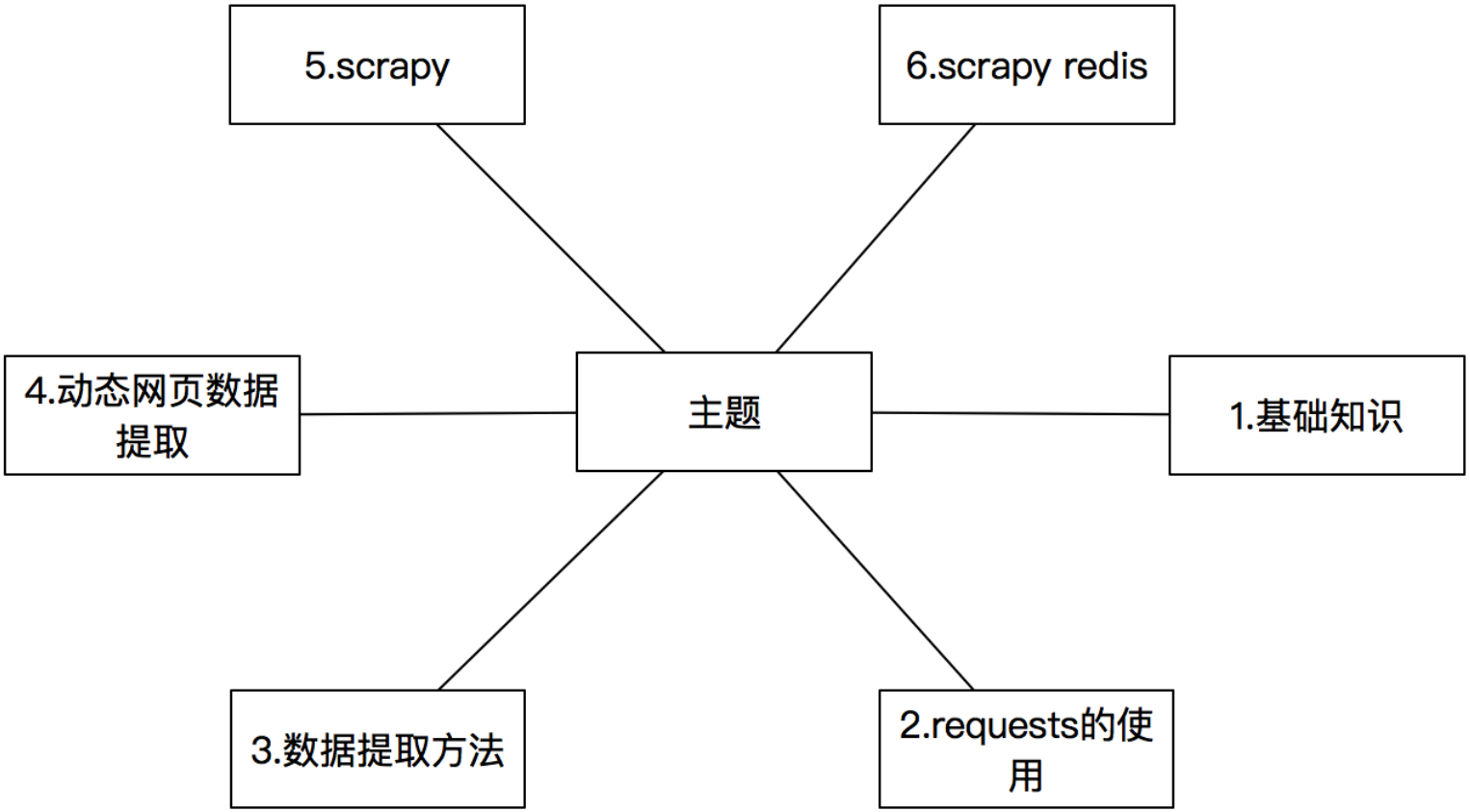
爬虫原理和数据抓取

课程概要


5.scrapy
6.scrapy redis
4.动态网页数据提取
主题
1.基础知识
3.数据提取方法
2.requests的使用

第一天课程概要

1、爬虫基础知识

2、HTTP和HTTPS的复习

3、字符串的复习

4、Requests的使用

5、Fiddle软件的使用

爬虫概要

1、爬虫的应用场景

2、爬虫的概念

3、爬虫的分类

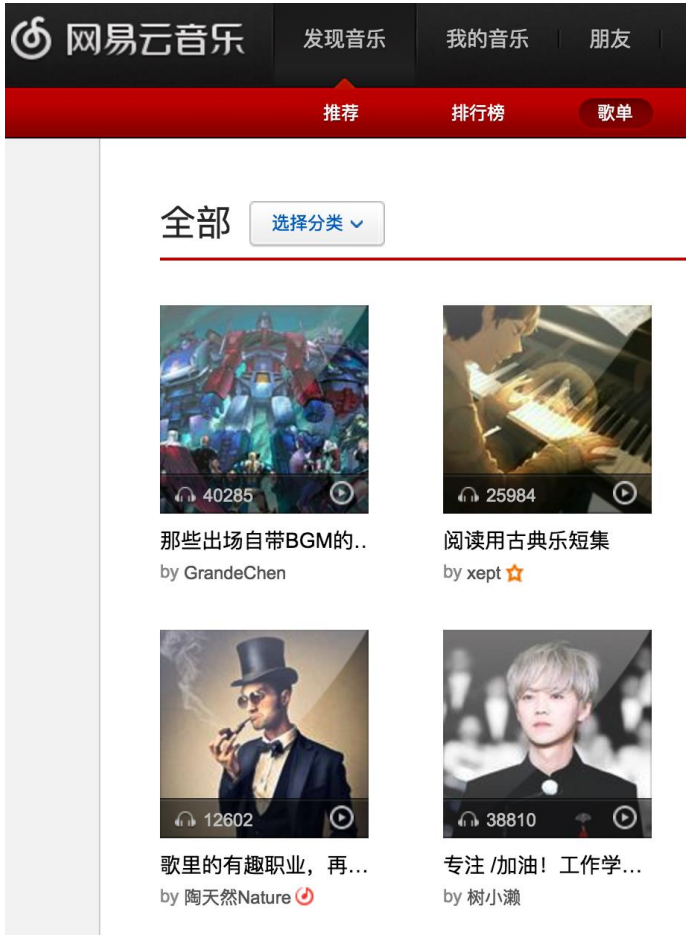
4、爬虫的工作流程

为什么要学习爬虫


微博
大家正在搜：好综艺好时光
首页
视频
发现
游戏
注册
登录
微指数 V1.0
nba总决赛
nba总决赛
输入关键词搜搜看
输入关键词搜搜看
选择日期
至
选择日期
热词趋势
实时趋势
地域解读
属性分析
热词趋势 2017-05-16 至 2017-06-15
整体趋势
PC&移动趋势
nba总决赛
大数据时代，我们的数据从哪里来
360K
300K
240K
180K
120K
60K
0
2017-05-16
2017-05-20
2017-05-24
2017-05-28
2017-06-01
2017-06-05
2017-06-09
2017-06-13

为什么要学习爬虫


Baidu 新闻
新闻全文
新闻标题
首页
百家号
国内
国际
军事
社会
财经
娱乐
热点要闻
个性推荐
进入推荐版
2020告别贫困 习近平要求限时完成的目标
习近平会见国际足联主席:中国重视足球运动 砥砺奋进的五年
绿色金融为何先在地方试验 听听总理怎么说
全国人大常委会举行委员长会议 俞正声会见巴基斯坦参联会主席
收入分配改革有变化 国家激励这些人增收
启动专项激励计划等三项试点 技能人才等7大群体有望增收
高温暴雨预警齐发 气象版图现"水火两重天"
北京交通整治：司机不礼让斑马线罚200元扣3分
中央和国家机关：今年率先实现生活垃圾强制分类
多地披露高考阅卷细节 有地区为数学题列10种解法
手机长途漫游费将取消 流量为啥还分本地和全国通用?
公安部长:严防黑恶势力染指党政机关和农村基层政权


网易云音乐
发现音乐
我的音乐
朋友
推荐
排行榜
歌单
全部
选择分类
40285
那些出场自带BGM的..
by GrandeChen
25984
阅读用古典乐短集
by xept
12602
歌里的有趣职业，再...
by 陶天然Nature
38810
专注 /加油！工作学...
by 树小濑

爬虫的定义

网络爬虫（又被称为网页蜘蛛，网络机器人）就是模拟客户端发送网络请求，接收请求响应，一种按照一定的规则，自动地抓取互联网信息的程序。

只要是浏览器能做的事情，原则上，爬虫都能够做

爬虫的更多用途

- 12306抢票
- 网站上的投票
- 短信轰炸

爬虫的分类

- **通用爬虫**： 通常指搜索引擎的爬虫
- **聚焦爬虫**： 针对特定网站的爬虫

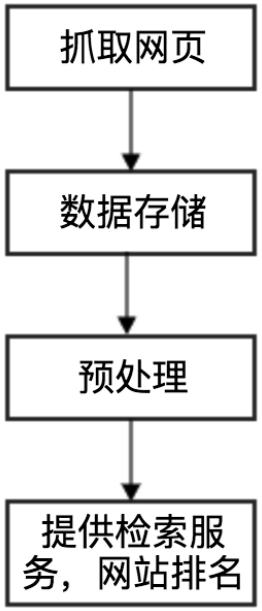
通用搜索引擎工作原理

想一想：

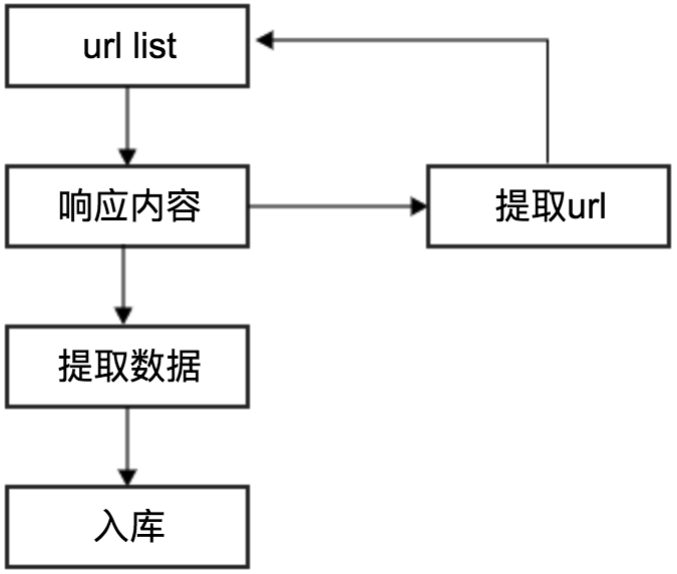
如果自己要实现一个和百度新闻一样的网站需要怎么做

通用爬虫和聚焦爬虫工作流程

搜索引擎流程



聚焦爬虫流程



通用搜索引擎的局限性

- 通用搜索引擎所返回的网页里90%的内容无用。
- 图片、音频、视频多媒体的内容通用搜索引擎无能为力
- 不同用户搜索的目的不全相同，但是返回内容相同

ROBOTS协议

Robots协议：网站通过Robots协议告诉搜索引擎哪些页面可以抓取，哪些页面不能抓取。

例如：https://www.taobao.com/robots.txt

HTTP和HTTPS复习内容

1、HTTP和HTTPS

2、HTTP的请求过程

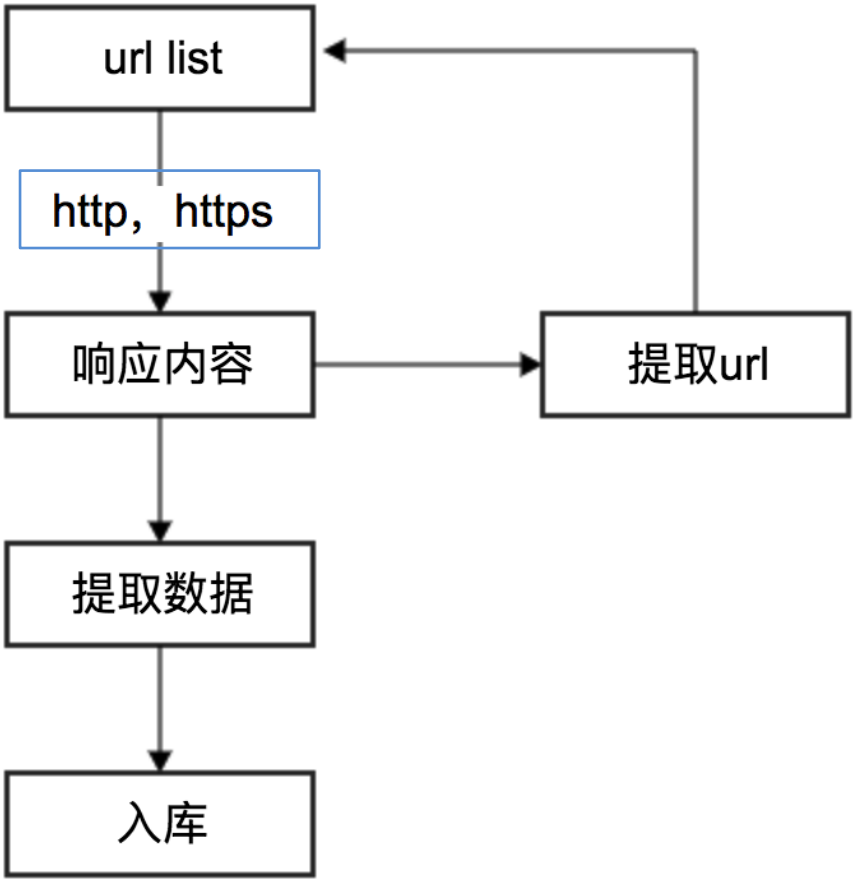
3、HTTP的请求形式

4、HTTP的常见请求头

5、GET和POST

6、响应状态码

复习HTTP HTTPS

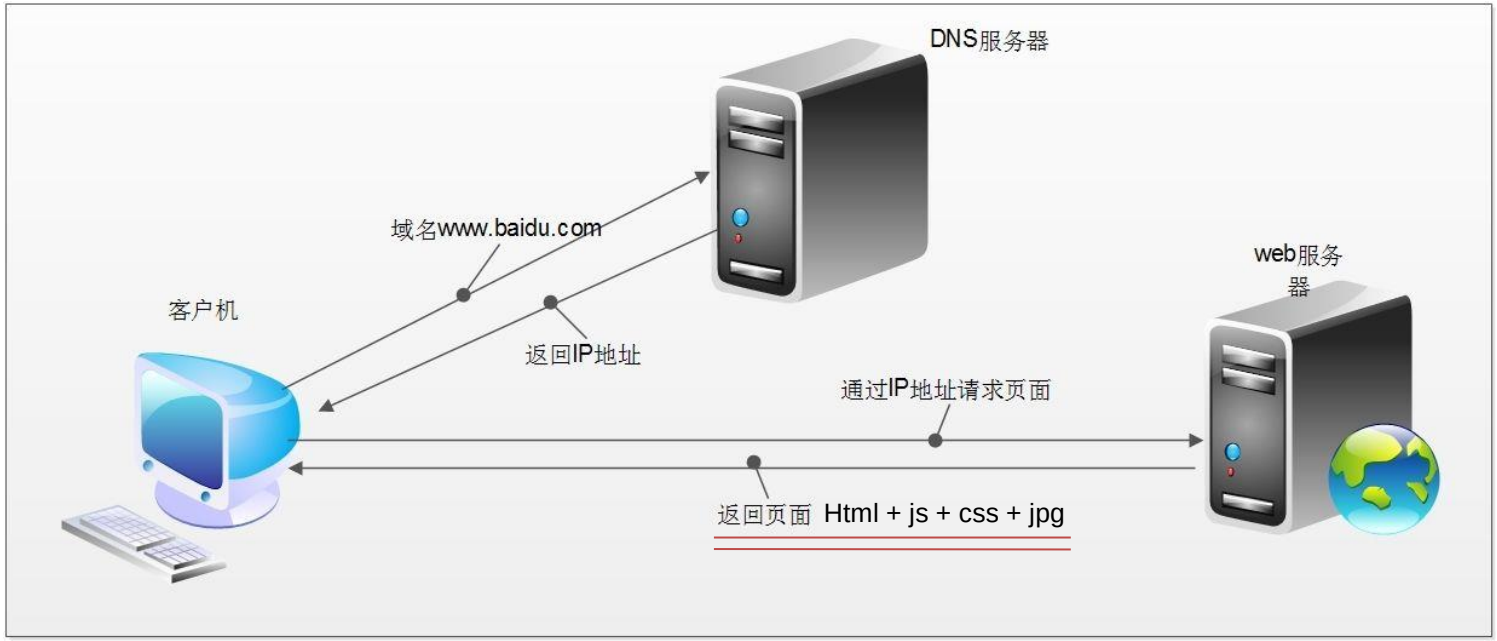


HTTP和HTTPS

- HTTP
  - 超文本传输协议
  - 默认端口号:80
- HTTPS
  - HTTP + SSL(安全套接字层)
  - 默认端口号：443

HTTPS比HTTP更安全，但是性能更低

浏览器发送HTTP请求的过程



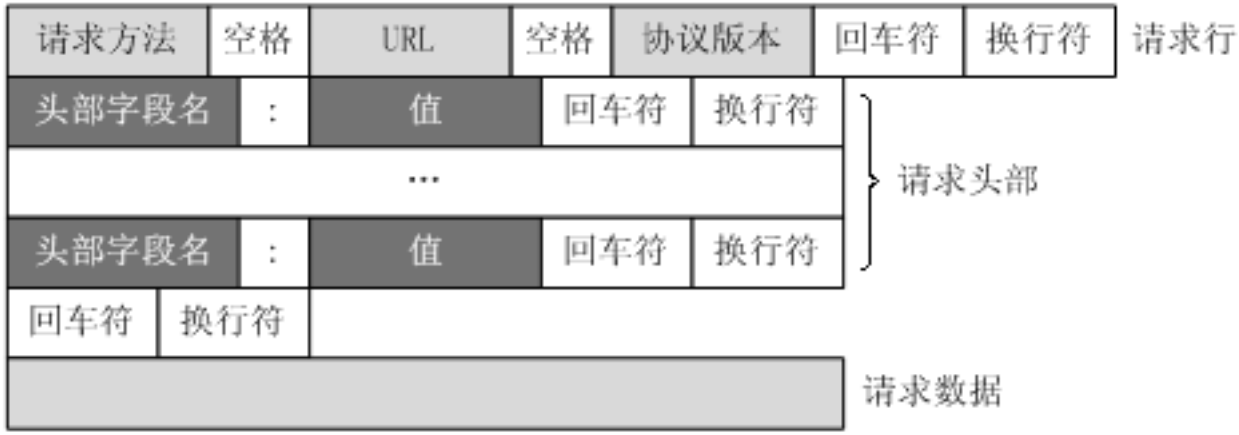
浏览器渲染出来的页面和爬虫请求的页面并不一样

url的形式

形式　scheme://host[:port#]/path/.../[?query-string][#anchor]

scheme：协议(例如：http, https, ftp)
host：服务器的IP地址或者域名
port：服务器的端口（如果是走协议默认端口，80 or 443）
path：访问资源的路径
query-string：参数，发送给http服务器的数据
anchor：锚（跳转到网页的指定锚点位置）
http://localhost:4000/file/part01/1.2.html
http://item.jd.com/11936238.html#product-detail

HTTP请求的形式


请求方法
空格
URL
空格
协议版本
回车符
换行符
请求行
头部字段名
:
值
回车符
换行符
...
请求头部
头部字段名
:
值
回车符
换行符
回车符
换行符
请求数据

HTTP常见请求头

1. Host (主机和端口号)
2. Connection (链接类型)
3. Upgrade-Insecure-Requests (升级为HTTPS请求)
4. User-Agent (浏览器名称)
5. Accept (传输文件类型)
6. Referer (页面跳转处)
7. Accept-Encoding（文件编解码格式）
8. Cookie （Cookie）
9. x-requested-with :XMLHttpRequest （是Ajax 异步请求)

常见的请求方法

- GET
- POST

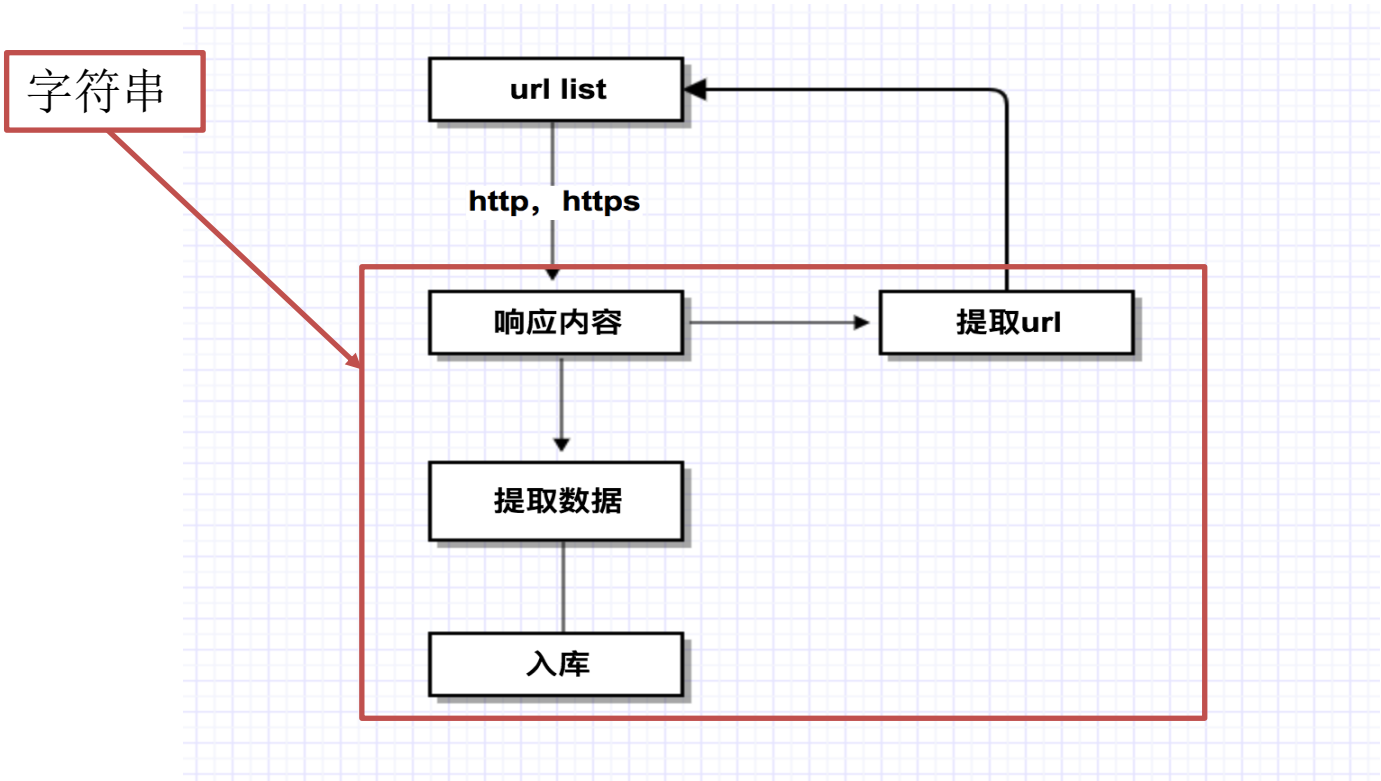
问题：GET方法和POST方法有什么区别呢？

响应状态码(status code)

- 200：成功
- 302：临时转移至新的url
- 307：临时转移至新的url
- 404：not found
- 500：服务器内部错误

字符串类型的区别和转化

问题：为什么要掌握python3字符串的相关知识



str类型和bytes类型

- bytes：二进制
  - 互联网上数据的都是以二进制的方式传输的
- str ： unicode的呈现形式

Unicode UTF8 ASCII的补充

字符(Character)是各种文字和符号的总称，包括各国家文字、标点符号、图形符号、数字等

字符集(Character set)是多个字符的集合

字符集包括：ASCII字符集、GB2312字符集、GB18030字符集、Unicode字符集等

ASCII编码是1个字节，而Unicode编码通常是2个字节。

UTF-8是Unicode的实现方式之一，UTF-8是它是一种变长的编码方式，可以是1，2，3个字节

str bytes如何转化

- str 使用encode方法转化为 bytes
- bytes通过decode转化为str
- 编码方式解码方式必须一样，否则就会出现乱码

Requests 使用入门

问题：为什么要学习requests，而不是urllib?

1. requests的底层实现就是urllib
2. requests在python2 和python3中通用，方法完全一样
3. requests简单易用
4. Requests能够自动帮助我们解压(gzip压缩的等)网页内容

requests的作用

作用：发送网络请求，返回响应数据

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

需要解决的问题：如何使用requests来发送网络请求?

发送简单的请求

需求：通过requests向百度首页发送请求，获取百度首页的数据

response = requests.get(url)

response的常用方法：

- response.text
- respones.content
- response.status_code
- response.request.headers
- response.headers

response.text 和response.content的区别

- response.text
  - 类型：str
  - 解码类型： 根据HTTP 头部对响应的编码作出有根据的推测，推测的文本编码
  - 如何修改编码方式：response.encoding=”gbk”
- response.content
  - 类型：bytes
  - 解码类型： 没有指定
  - 如何修改编码方式：response.content.deocde(“utf8”)

更推荐使用response.content.deocde()的方式获取响应的html页面

发送带header的请求

为什么请求需要带上header？

模拟浏览器，欺骗服务器，获取和浏览器一致的内容

- header的形式：字典
- headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36"}
- 用法： requests.get(url,headers=headers)

发送带参数的请求

什么叫做请求参数：

列1： http://www.webkaka.com/tutorial/server/2015/021013/ X

例2： https://www.baidu.com/s?wd=python&c=b

- 参数的形式：字典
- kw = {'wd':'长城'}
- 用法：requests.get(url,params=kw)

动手尝试

1、获取新浪首页，查看response.text 和response.content.decode()的区别

2、实现任意贴吧的爬虫，保存网页到本地

Requests深入

1. **发送POST请求**
2. **使用代理**
3. **处理cookies session**

发送POST请求

哪些地方我们会用到POST请求：

- 登录注册（ POST 比 GET 更安全）
- 需要传输大文本内容的时候（ POST 请求对数据长度没有要求）

所以同样的，我们的爬虫也需要在这两个地方回去模拟浏览器发送post请求

发送POST请求

用法：

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

data 的形式：字典

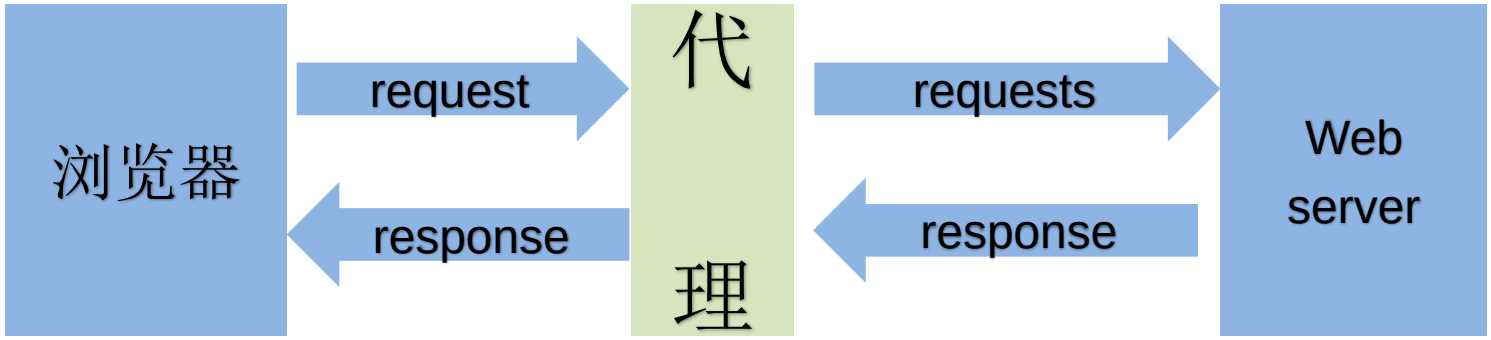
下面我们通过百度翻译的例子看看post请求如何使用

使用代理

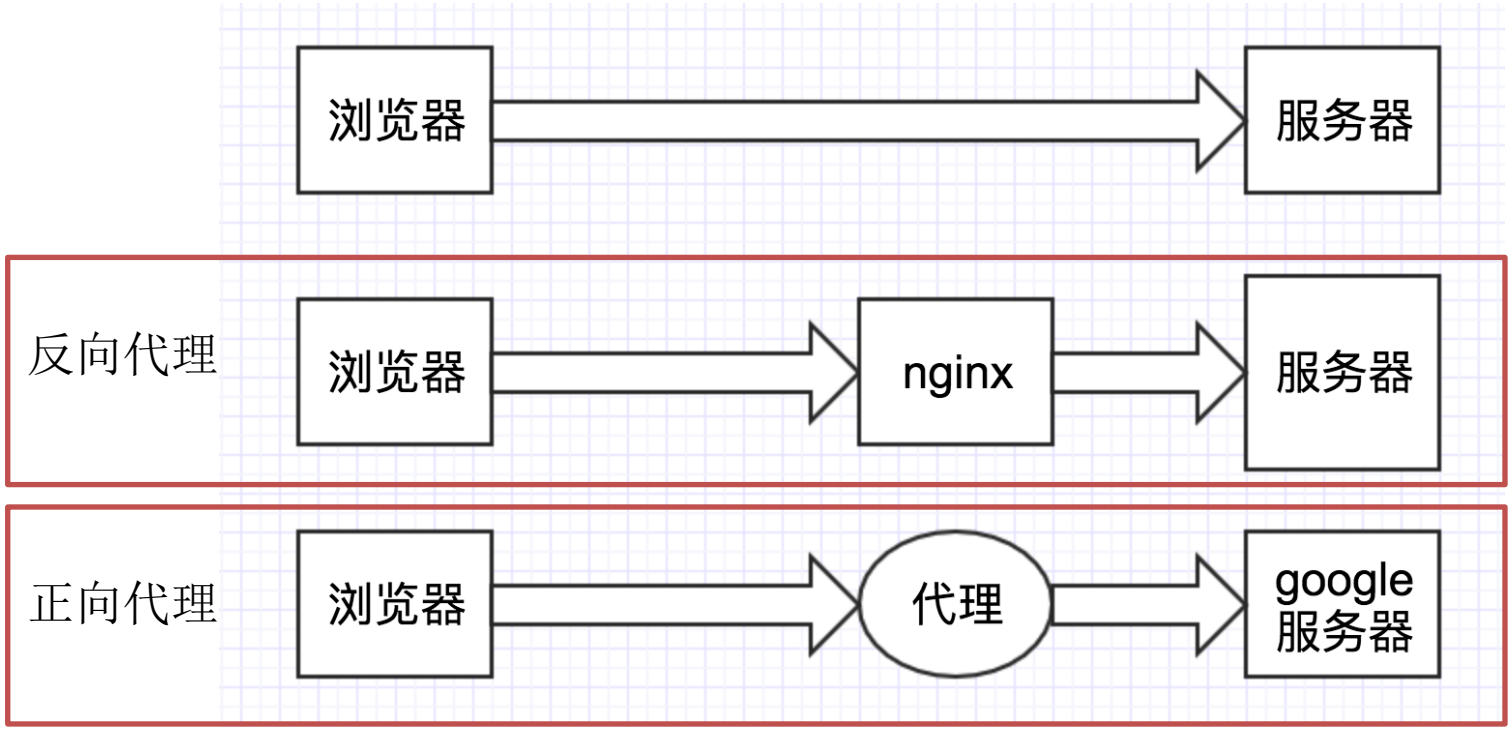
问题：为什么爬虫需要使用代理？

- 让服务器以为不是同一个客户端在请求
- 防止我们的真实地址被泄露，防止被追究

使用代理



使用代理



使用代理

用法：requests.get("http://www.baidu.com", proxies = proxies)

proxies的形式：字典

```
proxies = {
      "http": "http://12.34.56.79:9527",
      "https": "https://12.34.56.79:9527",
      }
```

cookie和session区别：

- cookie数据存放在客户的浏览器上，session数据放在服务器上。
- cookie不是很安全，别人可以分析存放在本地的cookie并进行cookie欺骗。
- session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能。
- 单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。

爬虫处理cookie和session

带上cookie、session的好处：

能够请求到登录之后的页面

带上cookie、session的弊端：

一套cookie和session往往和一个用户对应

请求太快，请求次数太多，容易被服务器识别为爬虫

不需要cookie的时候尽量不去使用cookie

但是为了获取登录之后的页面，我们必须发送带有cookies的请求

处理cookies 、session请求

requests 提供了一个叫做session类，来实现客户端和服务端的会话保持

使用方法：

1. 实例化一个session对象
2. 让session发送get或者post请求

```
session = requests.session()
response = session.get(url,headers)
```

动手尝试使用session来登录人人网：

http://www.renren.com/PLogin.do

Requests小技巧

1、reqeusts.util.dict_from_cookiejar 把cookie对象转化为字典

1.1. requests.get(url,cookies={})

2、**请求 SSL证书验证**

response = requests.get("https://www.12306.cn/mormhweb/ ", verify=False)

3、设置超时

response = requests.get(url,1)

4、配合状态码判断是否请求成功

assert response.status_code == 200

下面我们通过一个例子整体来看一下以上4点的用法

代理神器Fiddler

- 抓包工具
- Fiddler是一款强大Web调试工具，它能记录所有客户端和服务器的HTTP,HTTPS请求

Thank You!

改变中国 IT 教育，我们正在行动