卧槽666

Python中一些可能会问到的面试题

同步与异步

- 同步和异步关注的是消息通信机制 (synchronous communication/ asynchronous communication)
  所谓同步，就是在发出一个*调用*时，在没有得到结果之前，该*调用*就不返回。但是一旦调用返回，就得到返回值了。
  换句话说，就是由*调用者*主动等待这个*调用*的结果。
- 而异步则是相反，*调用*在发出之后，这个调用就直接返回了，所以没有返回结果。换句话说，当一个异步过程调用发出后，调用者不会立刻得到结果。而是在*调用*发出后，*被调用者*通过状态、通知来通知调用者，或通过回调函数处理这个调用。

谈谈python的装饰器，迭代器，yield

- 装饰器:装饰器的本质是一个闭包函数,他的作用就是让其他函数在不需要做任何代码修改的前提下增加额外功能,装饰器的返回值也是一个函数对象.我们通常在一些有切面需求的场景,比如:插入日志,性能测试,事务处理,缓存,权限校验等场景,有了装饰器我们就可以少写很多重复代码,提高工作效率.
- 迭代器:迭代器是一中访问可迭代对象的方式,通常从第一个元素开始访问,知道所有的元素都被访问完才结束,迭代器只能前进不能后退,使用迭代器可以不用事先准备好带跌过程中的所有元素,仅仅是在迭代到该元素的时候才计算该元素,而在这之前的元素则是被销毁,因此迭代器适合遍历一些数据量巨大的无限的序列.
  迭代器的本质就是调用__iter__方法,每次调用的时候返回一个元素,当没有下一个元素的时候会抛出StopIteration异常

python适合的场景有哪些？当遇到计算密集型任务怎么办?

- 适用场景:网站运维,金融分析,服务器编写,爬虫
- 当遇到io密集型任务时,涉及到的大多是网络,磁盘等任务,这一类任务的特性是cpu小号低,使用多线程.
- 计算密集型的任务主要是消耗cpu性能,谁要运用多进程,当然运用python语言的运行效率很低,所以一般对于计算密集型任务,可以使用c语言编写

谈谈mysql字符集和排序规则

- 字符集，即用于定义字符在数据库中的编码的集合。 常见的字符集：utf-8 gbk 等
- 排序规则，就是指字符比较时是否区分大小写，以及是按照字符编码进行比较还是直接用二进制数据比较。

说一下线程、进程、协程?

- 回答：进程是具有一定独立功能的程序关于某个数据集合上的一 次运行活动,进程是系统进行资源分配和调度的一个独立单位。
- 每个进程都有自己的独立内存空间，不同进程通过进程间通信 来通信。由于进程比较重量，占据独立的内存，所以上下文进程间的切换开销(栈、寄存器、虚拟内存、文件句柄等)比较 大，但相对比较稳定安全。
- 线程是进程的一个实体,是 CPU 调度和分派的基本单位, 它是比进程更小的能独立运行的基本单位.线程自己基本上不 拥有系统资源,只拥有一点在运行中必不可少的资源(如程序计 数器,一组寄存器和栈),但是它可与同属一个进程的其他的线 程共享进程所拥有的全部资源。线程间通信主要通过共享内存， 上下文切换很快，资源开销较少，但相比进程不够稳定容易丢 失数据。
- 协程是一种用户态的轻量级线程，协程的调度完全由用户 控制。协程拥有自己的寄存器上下文和栈。协程调度切换时,将寄存器上下文和栈保存到其他地方，在切回来的时候，恢复先前保存的寄存器上下文和栈，直接操作栈则基本没有内核切 换的开销，可以不加锁的访问全局变量，所以上下文的切换非 常快。
- 最好说一下在项目中如何使用，举个例子

如何解决线程安全?

- 线程安全是在多线程的环境下，能够保证多个线程同时执行时程序依旧运行正确, 而且要保证对于共享的数据可以由多个线程存取，但是同一时刻只能有一个线程进行存取。多线程环境下解决资源竞争问题的办法是加锁来保证存取操作的唯一性。如何加锁？ 分布

式 负载均衡

常用的linux命令

- ls,cd,more,clear,mkdir,pwd,rm,grep,find,mv,su,date等等

什么是面向对象编程?

- 答技巧:说一下什么事面向对象编程，再说说为什么要面向对象编程，面向对象特性。
- 回答：
  面向对象编程是一种解决软件复用的设计和编程方法。 这种方法把软件系统中相近相似的操作逻辑和操作应用数据、状态,以类的型式 述出来,以对象实例的形式在软件系统中复用,以达到高软件开发效率的作用。封装,继承多态.

如何提高 Python 的运行效率，请说出不少于 2 种提高运行效率的方法?

- 1.使用生成器
- 2.关键代码使用外部功能包:Cython、Pylnlne、PyPy、Pyrex
- 3.针对循环的优化——尽量避免在循环中访问变量的属性;

说一下 MySQL 数据库存储的原理?

- 回答技巧:先回答一下mysql的原理，拓展一下它的有点，或者mysql你是怎么用的?
- 回答：储存过程是一个可编程的函数，它在数据库中创建并保存。它可以有 SQL 语句和一些特殊的控制结构组成。当希望在不同的应用程序或平台上执行相同的函数，或者封装特定功能时，存储过程是非常有用的。数据库中的存储过程可以看做是对编程中面向对象方法的模拟。它允许控制数据的访问方式。 存储过程通常有以下优点:
- 1)存储过程能实现较快的执行速度。
- 2)存储过程允许标准组件是编程。
- 3)存储过程可以用流控制语句编写，有很强的灵活性，可 以完成复杂的判断和较复杂的运算。
- 4)存储过程可被作为一种安全机制来充分利用。
- 5)存储过程能过减少网络流量

你工作中遇到哪些bug,怎么解决的?

- 回答技巧:不要聊一些小bug，聊一些印象深刻。或者可以把这道题转化成你遇到什么困难
- 回答:
  - 1.刚入行的时候，对业务不太熟悉，加上给的业务文档不清晰，导致加班很多。或者第一次代码合并，python版本更新带来问题等等
  - 2.项目中第一次做登录模块/支付模块，不太熟悉，吃了很多苦头
  - 3.平时敲代码中积累的bug

说一下事务的特性?

- 1．原子性(Atomicity):事务中的全部操作在数据库 中是不可分割的，要么全部完成，要么均不执行。
- 2．一致性(Consistency):几个并行执行的事务，其执行 结果必须与按某一顺序串行执行的结果相一致。
- 3．隔离性(Isolation):事务的执行不受其他事务的干扰， 事务执行的中间结果对其他事务必须是透明的。
- 4、持久性(Durability):对于任意已 交事务，系统必须 保证该事务对数据库的改变不被丢失，即使数据库出

redis 和 mysql 的区别?

- 回答技巧:回答题目有什么区别，然后其中一个举例子，你是如何使用?
- 回答:
  - redis 是内存数据库，数据保存在内存中，速度快。
  - mysql 是关系型数据库，持久化存储，存放在磁盘里面， 功能强大。检索的话，会涉及到一定的 IO，数据访问也就慢。
  - 我比较常用是mysql，主要创建数据库，创建表，数据操作，增删改查，我做的比较多是查，例如在xxx项目中，有个搜索模块，当时我做比较简单就是用模糊匹配去做搜索。

redis 受攻击怎么办?

- 回答：
  - 在工作防止redis受攻击，我都会做一下措施：
  - 1.主从
  - 2.持久化存储Redis 不以 root 账户启动
  - 3.设置复杂密码
  - 4.不允许 key 方式登录

说一下mongoDB你是如何使用?

- 回答技巧:回答mongoDB什么,优缺点，平时怎么用的?
  - MongoDB是一个面向文档的数据库系统。使用C++编写，不 支持 SQL，但有自己功能强大的查询语法。

2. Re:Python中一些可能

非常感谢楼主，楼主好，

阅读排行榜

1. Python中一些可能会
8)

评论排行榜

1. Python中一些可能会

推荐排行榜

1. Python中一些可能会

- MongoDB 使用 BSON 作为数据存储和传输的格式。BSON 是 一种类似 JSON 的、二进制序列化文档，支持嵌套对象和数组。
- MongoDB 很像 MySQL，document 对应 MySQL 的 row， collection 对应- MySQL 的 table
- 缺点:不支持事务，MongoDB 占用空间过大，维护工具不够成熟

- 应用场景:
  - 1.网站数据:mongo 非常适合实时的插入，更新与查询， 并具备网站实时数据存储所需的复制及高度伸缩性。
  - 2.缓存:由于性能很高，mongo 也适合作为信息基础设 施的缓存层。在系统重启之后，由 mongo 搭建的持久化缓存可 以避免下层的数据源过载。
  - 3.大尺寸、低价值的数据:使用传统的关系数据库存储 一些数据时可能会比较贵，在此之前，很多程序员往往会选择 传统的文件进行存储。
  - 4.高伸缩性的场景:mongo 非常适合由数十或者数百台 服务器组成的数据库。
  - 5.用于对象及 JSON 数据的存储:mongo 的 BSON 数据格 式非常适合文档格式化的存储及查询。
  - 6.重要数据:mysql，一般数据:mongodb，临时数据: memcache

Redis与mongodb的优缺点?

- 回答技巧:先说一下两者区别，再说它们优缺点
  - 回答: MongoDB 和 Redis 都是 NoSQL，采用结构型数据存储。二者在使用场景中，存在一定的区别，这也主要由于二者在内存映射的处理过程，持久化的处理方法不同。MongoDB 建议集群部署，更多的考虑到集群方案，Redis 更偏重于进程顺序写入， 虽然支持集群，也仅限于主-从模式.

- Redis 优点:
  - 1 读写性能优异
  - 2 支持数据持久化，支持 AOF 和 RDB 两种持久化方式
  - 3 支持主从复制，主机会自动将数据同步到从机，可以进 行读写分离。
  - 4 数据结构丰富:除了支持 string 类型的 value 外还支 持 string、hash、set、sortedset、list 等数据结构。

- 缺点:
  - 1 Redis 不具备自动容错和恢复功能，主机从机的宕机都 会导致前端部分读写请求失败，需要等待机器重启或者手动切 换前端的 IP 才能恢复。
  - 2 主机宕机，宕机前有部分数据未能及时同步到从机，切 换 IP 后还会引入数据不一致的问题，降低了系统的可用性。
  - Redis 较难支持在线扩容，在集群容量达到上限时在线 扩容会变得很复杂。为避免这一问题，运维人员在系统上线时必须确保有足够的空间，这对资源造成了很大的浪费.避免这一问题，运维人员在系统上线时 必须确保有足够的空间，这对资源造成了很大的浪费。

- mongodb的优缺点：
  - 优点:弱一致性(最终一致)，更能保证用户的访问速度 文档结构的存储方式，能够更便捷的获取数 频)
  - 内置 GridFS，高效存储二进制大对象 (比如照片和视
  - 支持复制集、主备、互为主备、自动分片等特性 动态查询 全索引支持,扩展到内部对象和内嵌数组
  - 缺点:不支持事务MongoDB 占用空间过大，维护工具不够成熟

数据库怎么优化查询效率?

- 回答技巧:有条理按照题目回答即可
- 回答：
  - 1.储存引擎选择:如果数据表需要事务处理，应该考虑使用 InnoDB，因为它完全符合ACID 特性。如果不需要事务处理，使用默认存储引擎MyISAM 是比较明智的
  - 2.分表分库，主从
  - 3.对查询进行优化，要尽量避免全表扫 ，首先应考虑在 where 及 order by 涉及的列上建立索引
  - 4.应尽量避免在 where 子句中对字段进行 null 值判断， 否则将导致引擎放弃使用索引而进行全表扫
  - 5.应尽量避免在 where 子句中使用 != 或 <> 操作符， 否则将引擎放弃使用索引而进行全表扫
  - 6.应尽量避免在 where 子句中使用 or 来连接条件，如果 一个字段有索引，一个字段没有索引，将导致引擎放弃使用索 引而进行全表扫
  - 7.Update语句，如果只更改1、2 个字段，不要Update全部字段，否则频繁调用会引起明显的性能消耗，同时带来大 量日志

数据库优化方案?

- 回答技巧:如题回答即可
  - 1.优化索引、SQL 语句、分析慢查询;
  - 2.设计表的时候严格根据数据库的设计范式来设计数据库;
  - 3.使用缓存，把经常访问到的数据而且不需要经常变化的 数据放在缓存中，能节约磁盘 IO;
  - 4.优化硬件;采用 SSD，使用磁盘队列技术 (RAID0,RAID1,RDID5)等;
  - 5.采用 MySQL 内部自带的表分区技术，把数据分层不同 的文件，能够 高磁盘的读取效率;
  - 6.垂直分表;把一些不经常读的数据放在一张表里，节约 磁盘 I/O;

- 7.主从分离读写;采用主从复制把数据库的读操作和写入 操作分离开来;
- 8.分库分表分机器(数据量特别大)，主要的的原理就是数据路由;
- 9.选择合适的表引擎，参数上的优化;
- 10.进行架构级别的缓存，静态化和分布式;
- 11.不采用全文索引;
- 12.采用更快的存储方式，例如 NoSQL 存储经常访问的数

redis 基本类型、相关方法?

- 回答技巧:先回答问题所问的，找其中一个说明一下如何使用，在拓展一下redis如何使用的
- 回答: Redis支持五种数据类型:string(字符串)、hash(哈希)、list(列表)、set (集合)及zset(sorted set:有序集合)。
  - String是Redis最为常用的一种数据类型，String的数据结构为key/value类型，String 可以包含任何数据。常用命令: set,get,decr,incr,mget 等

redis 的使用场景有哪些?

- 回答技巧:优点， 如题回答即可
- 回答:
  -1.取最新 N 个数据的操作
  -2.排行榜应用,取 TOP N 操作
  -3.需要精准设定过期时间的应用
  -4.计数器应用
  -5.uniq 操作,获取某段时间所有数据排重值
  -6.Pub/Sub 构建实时消息系统
  -7.构建队列系统
  -8.缓存

说一下冒泡排序?

- 回答技巧:回答冒泡原理，最好能手写，拓展一下其他排序?
- 冒泡排序的思想: 每次比较两个相邻的元素, 如果他们的顺序错误就把他们交换位置。

```
def bubble_improve(l): print l
flag = 1
for index in range(len(l) - 1, 0 , -1):
if flag:
flag = 0 for two_index in range(index):
if l[two_index] > l[two_index + 1]:
l[two_index], l[two_index + 1] = l[two_index + 1], l[two_index]
flag = 1
else: break
print ll = [10, 20, 40, 50, 30, 60] bubble_improve(l)
```

说一下 Django，MIDDLEWARES 中间件的作用?

- 回答技巧:如题回答即可。
- 回答: 中间件是介于 request 与 response 处理之间的一道 处理过程，相对比较轻量级，并且在全局上改变 django 的输入与输出。

说一下mvvm ?

- 回答技巧:说一下MVVM，然后拓展回熟悉的MVT ?
- MVVM:将“数据模型数据双向绑定”的思想作为核心，在 View 和 Model 之间没有联系，通过 ViewModel进行交互，而且 Model 和 ViewModel之间的交互是双向的，因此视图的数据的变化会同时修改数据源，而数据源数据的变化也会立即反应到View上。

你对 Django 的认识?

- 回答技巧:说一下Django是什么，然后说一下它的优缺点，再说说项目怎么用?
- 回答: Django 是走大而全的方向，它最出名的是其全自动化的管 理后台:只需要使用起 ORM，做简单的对象定义，它就能自动生成数据库结构、以及全功能的管理后台。
  - 优点:
    超高的开发效率。
    适用的是中小型的网站，或者是作为大型网站快速，实现产品雏形的工具。
    彻底的将代码、样式分离; Django 从根本上杜绝在模板中进行编码、处理数据的可能。
  - 缺点:
    其性能扩展有限;
    采用 Django 的项目，在流量达到一定规模后，都需要对其进行重构，才能满足性能的要求。
    Django 内置的 ORM 跟框架内的其他模块耦合程度高。

说一下Jieba 分词?

- 回答技巧:jieba分词有哪些，作用是什么?

- 回答：
  - Jieba 分词支持三种分词模式:
  - 精确模式：试图将句子最精确地切开，适合文本分析;
  - 全局模式：把句子中所有的可以成词的词语都扫出来, 速度非常快，但是不能解决歧义;
  - 搜索引擎模式：在精确模式的基础上，对长词再次切分，高召回率，适合用于搜索引擎分词
  - 功能:
  - 分词，添加自定义词典，关键词取，词性标注，并行分词，Tokenize:返回词语在原文的起始位置，ChineseAnalyzer for Whoosh 搜索引擎

    Django 重定向你是如何实现的?用的什么状态码?

- 回答技巧:如题回答即可
- 使用 HttpResponseRedirect redirect 和 reverse ，状态码:302,301

爬取下来的数据如何去重，说一下具体的算法依据?

- 回答技巧:如题回答即可
- 答:
  - 1.通过 MD5 生成电子指纹来判断页面是否改变
  - 2.nutch 去重。nutch 中 digest 是对采集的每一个网页内 容的 32 位哈希值，如果两个网页内容完全一样，它们的 digest 值肯定会一样。

    写爬虫是用多进程好?还是多线程好? 为什么?

- 回答技巧:就是对比多线程与多进程爬虫的优缺点
- 回答：IO 密集型代码(文件处理、网络爬虫等)，多线程能够有效升效率(单线程下有 IO 操作会进行 IO 等待，造成不必要的时间浪费，而开启多线程能在线程 A 等待时，自动切换到线程B可以不浪费CPU的资源，从而能升程序执行效率)。在实际的数据采集过程中，既考虑网速和响应的问题，也需要考虑自身机器的硬件情况，来设置多进程或多线程。

1.说一下 numpy 和 pandas 的区别?分别的应用场景?

- 回答技巧:如题回答就好
- Numpy 是数值计算的扩展包，纯数学。
- ePandas做数据处理以矩阵为基础的数学计算模块。供了一套名为 DataFrame 的数据结构，比较契合统计分析中的表结构，并且供了计算接口，可用Numpy 或其它方式进行计算。

验证码如何处理?

- 回答技巧:如题回答
  - 1．Scrapy 自带处理验证码
  - 2．获取到验证码图片的url，调用第三方付费借口破解验证码

    动态的股票信息如何抓取?

- 回答技巧:先说一下抓取方法，然后举个例子
- 股票数据的获取目前有如下两种方法可以获取:
  - 1.http/JavaScript 接口取数据
  - 2.web-service 接口
  - Sina 股票数据接口
  - 以大秦铁路(股票代码:601006)为例，如果要获取它的 最新行情，只需访问新浪的股票数据
  - 接口:http://hq.sinajs.cn/list=sh601006 这个 url 会返回一串文本，例如 var hq_str_sh601006="大秦铁路, 27.55, 27.25, 26.91, 27.55, 26.20, 26.91, 26.92, 22114263, 589824680, 4695, 26.91, 57590, 26.90, 14700, 26.89, 14300, 26.88, 15100, 26.87, 3100, 26.92, 8900, 26.93, 14230, 26.94, 25150, 26.95, 15220, 26.96, 2008-01-11, 15:05:32";

    scrapy 去重?

- 回答技巧:从各个方面有条理去回答
- 数据量不大时，可以直接放在内存里面进行去重，python 可以使用 set()进行去重。
- 当去重数据需要持久化时可以使用 redis 的 set 数据结构。
- 当数据量再大一点时，可以用不同的加密算法先将长字符 串压缩成 16/32/40 个字符，再使用上面两种方法去重;
- 当数据量达到亿(甚至十亿、百亿)数量级时，内存有限， 必须用“位”来去重，才能够满足需求。Bloomfilter 就是将 去重对象映射到几个内存“位”，通过几个位的 0/1 值来判断一个对象是否已经存在。 然而 Bloomfilter 运行在一台机器的内存上，不方便持久化(机器 down 掉就什么都没啦)，也不方便分布式爬虫的统一去重。如果可以在 Redis 上申请内存进行 Bloomfilter，以上两个问题就都能解决了。
- simhash 最牛逼的一点就是将一个文档，最后转换成一个 64 位的字节，暂且称之为特征字，然后判断重复只需要判断他 们的特征字的距离是不是<n(根据经验这个 n 一般取值为 3)，就可以判断两个文档是否相似。

分布式有哪些方案，哪一种最好?

- 回答技巧:先说一下有什么方案，分析哪一个好?
- celery、beanstalk，gearman个人认为 gearman 比较好。
  原因主要有以下几点:
  - 1).技术类型简单，维护成本低。
  - 2).简单至上。能满足当前的技术需求即可 (分布式任务 处理、异步同步任务同时支持、任务队列的持久化、维护部署 简单)。
  - 3).有成熟的使用案例。instagram 就是使用的 gearman 来完成图片的处理的相关任务，有成功的经验，我们当然应该 借鉴。

Post 和 get 区别?

- 回答技巧:有条理从各方面去回答即可
  - 1、GET请求，请求的数据会附加在URL之后，以分割 URL 和传输数据，多个参数用&连接。URL的编码格式采用的是 ASCII 编码，而不是 uniclde，即是说所有的非ASCII字符都要编码之后再传输。
    POST 请求:POST请求会把请求的数据放置在HTTP请求包的包体中。上面的 item=bandsaw 就是实际的传输数据。因此，GET 请求的数据会暴露在地址栏中，而 POST 请求则不会。
  - 2、传输数据的大小
    在HTTP规范中，没有对 URL 的长度和传输的数据大小进行限制。但是在实际开发过程中，对于 GET，特定的浏览器和服务器对 URL 的长度有限制。因此，在使用GET请求时，传输数据会受到URL长度的限制。

谈一谈你对 Selenium 和 PhantomJS 了解?

- 回答技巧：如何回答即可
- Selenium 是一个 Web 的自动化测试工具，可以根据我们的指令，让 浏览器自动加载页面，获取需要的数据，甚至页面截屏，或者判断网 站上某些动作是否发生。Selenium 自己不带浏览器，不支持浏览器 的功能，它需要与第三方浏览器结合在一起才能使用。但是我们有时 候需要让它内嵌在代码中运行，所以我们可以用一个叫 PhantomJS 的工具代替真实的浏览器。Selenium 库里有个叫 WebDriver 的 API。 WebDriver 有点儿像可以加载网站的浏览器，但是它也可以像 BeautifulSoup 或者其他 Selector 对象一样用来查找页面元素，与 页面上的元素进行交互 (发送文本、点击等)，以及执行其他动作来 运行网络爬虫。
- PhantomJS 是一个基于Webkit的“无界面”(headless)浏览器， 它会把网站加载到内存并执行页面上的 JavaScript，因为不会展示 图形界面，所以运行起来比完整的浏览器要高效。
  如果我们把 Selenium 和 PhantomJS 结合在一起，就可以运行一 个非常强大的网络爬虫了，这个爬虫可以处理 JavaScrip、Cookie、 headers，以及任何我们真实用户需要做的事情。

常用的反爬虫措施?

- 回答技巧: 有条理从各方面去回答即可
- 回答: 1.添加代理 2.降低访问频率 3.User-Agent4. 动态 HTML 数据加载 5. 验证码处理6. Cookie

常见的反爬虫的应对方法?

- 回答技巧:有条理从各方面去回答即可
- 1).通过 Headers 反爬虫，从用户请求的 Headers 反爬虫是最常见的反爬虫策略。很多网站都 会对 Headers 的 User-Agent 进行检测，还有一部分网站会对 Referer 进行检测(一些资源网站的防盗链就是检测 Referer)。如果遇到了 这类反爬虫机制，可以直接在爬虫中添加 Headers，将浏览器的 User-Agent 复制到爬虫的 Headers 中;或者将 Referer 值修改为目 标网站域名。对于检测 Headers 的反爬虫，在爬虫中修改或者添加 Headers 就能很好的绕过。
- 2).基于用户行为反爬虫
  还有一部分网站是通过检测用户行为，例如同一IP 短时间内多次访问同一页面，或者同一账户短时间内多次进行相同操作。 大多数网站都是前一种情况，对于这种情况，使用 IP 代理就可以 解决。可以专门写一个爬虫，爬取网上公开的代理 ip，检测后全部 保存起来。这样的代理 ip 爬虫经常会用到，最好自己准备一个。有 了大量代理 ip 后可以每请求几次更换一个 ip，这在 requests或者 urllib2 中很容易做到，这样就能很容易的绕过第一种反爬虫。
  对于第二种情况，可以在每次请求后随机间隔几秒再进行下一次请 求。有些有逻辑漏洞的网站，可以通过请求几次，退出登录，重新登 录，继续请求来绕过同一账号短时间内不能多次进行相同请求的限制。
- 3).动态页面的反爬虫
  上述的几种情况大多都是出现在静态页面，还有一部分网站，我们 需要爬取的数据是通过 ajax 请求得到，或者通过 JavaScript 生成的。 首先用 Fiddler 对网络请求进行分析。如果能够找到 ajax 请求，也 能分析出具体的参数和响应的具体含义，我们就能采用上面的方法，直接利用 requests 或者 urllib2 模拟 ajax 请求，对响应的 json 进 行分析得到需要的数据。
- 能够直接模拟 ajax 请求获取数据固然是极好的，但是有些网站把 ajax 请求的所有参数全部加密了。我们根本没办法构造自己所需要 的数据的请求。这种情况下就用 selenium+phantomJS，调用浏览器 内核，并利用 phantomJS 执行 js 来模拟人为操作以及触发页面中的 js 脚本。从填写表单到点击按钮再到滚动页面，全部都可以模拟， 不考虑具体的请求和响应过程，只是完完整整的把人浏览页面获取数 据的过程模拟一遍。
- 用这套框架几乎能绕过大多数的反爬虫，因为它不是在伪装成浏览 器来获取数据(上述的通过添加 Headers 一定程度上就是为了伪装 成浏览器)，它本身就是浏览器，phantomJS 就是一个没有界面的浏览器，只是操控这个浏览器的不是人。利 selenium+phantomJS 能干很多事情，例如识别点触

分类: 基础学习

卧槽666
关注 - 1
粉丝 - 8
+加关注

3 0

posted @ 2017-12-26 21:13 卧槽666 阅读(4148) 评论(2) 编辑 收藏

评论列表

#1楼 2018-01-10 21:10 徐旭东

非常感谢楼主，楼主好人！

支持(0) 反对(0)

#2楼 2018-04-25 15:31 初学者_菜鸟

6666666666

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

相关博文：
· python中基于descriptor的一些概念（上）
· 一些可能没用过的调试窗口
· python面试题大全（一）
· Python中的Class
· 面试中经常问到的问题总结（1）

最新新闻：
· Airbnb因发布非法广告遭巴黎起诉 面临1400万美元罚金
· 仅用326天 《堡垒之夜》解锁5亿美元成就
· 微软提交专利申请 将柔性织物触控传感器放在Surface设备背面
· 四大银行贷款流向揭示了哪些真相?
· “跨国视频造假窝点”曝光！帮AI揪出99%换脸视频
» 更多新闻...