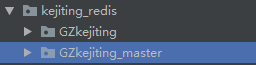
# 关于如何把目前单机爬虫改造成分布式爬虫操作步骤

在未来爬虫数量增加，为了减轻爬虫平台服务器的压力，需要将之前的单机爬虫改造成分布式爬虫。为了改造方便，最好是把之前的单机爬虫当做从爬虫，在新建一个新的爬虫项目（用作主爬虫）把单机爬虫详情页之前的代码复制到主爬虫中即可。

## 建立一个文件夹用于存放主从爬虫

例如科技厅的爬虫代码：建立一个名叫kejiting\_redis的文件夹存放主从爬虫项目。



## 主爬虫配置

* 在主爬虫的settings文件中添加

DUPEFILTER\_CLASS = "scrapy\_redis.dupefilter.RFPDupeFilter"

REDIS\_HOST = 'localhost'  
REDIS\_PORT = 6379

Redis\_host为redis数据库的ip，到时候修改为主服务器的ip即可。

由于主爬虫只是存储url, 目前不需要通过pipelines存储，因此item与pipelines都不需要配置。

* 主爬虫spider的改造

雪花算法：单机爬虫雪花算法是在spider的最前面就实例化了，由于分布式爬虫将主从爬虫分开，现将雪花算法的实例化改在从爬虫spider的最前面。

* Redis数据库的连接

在spider的类中定义如下连接数据库函数（最好是在一开始就定义）

def \_\_init\_\_(self):

self.redis\_url = 'redis://localhost:6379/'

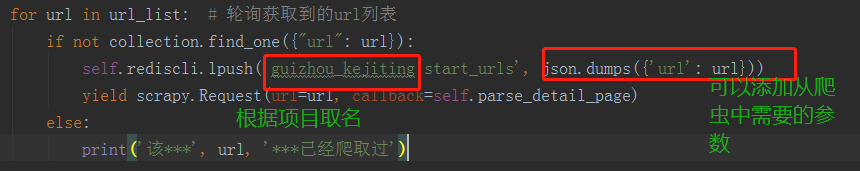
self.rediscli = redis.Redis(host='localhost', port=6379, db=0)

* url的存储

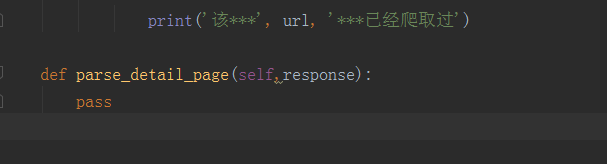
在获取到url后添加存储url的代码即可

self.rediscli.lpush('guizhou\_kejiting:start\_urls',

json.dumps({'url': url}))



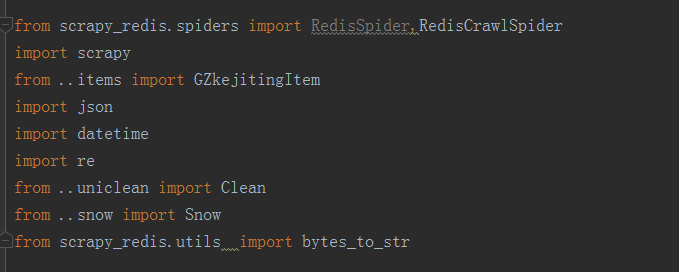
* 注意后面的callback函数pass就行



## 从爬虫配置

建议把之前的单机爬虫作为从爬虫就不要在配置数据存储过程：item、pipelines都不需要修改。

* 需要import的包（uniclean是我自己的数据清理的函数）



* Settings配置

SCHEDULER = "scrapy\_redis.scheduler.Scheduler"

DUPEFILTER\_CLASS = "scrapy\_redis.dupefilter.RFPDupeFilter"

REDIS\_URL = 'redis://127.0.0.1:6379'

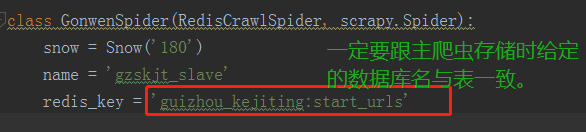
ip后期需要修改

* 从爬虫spider

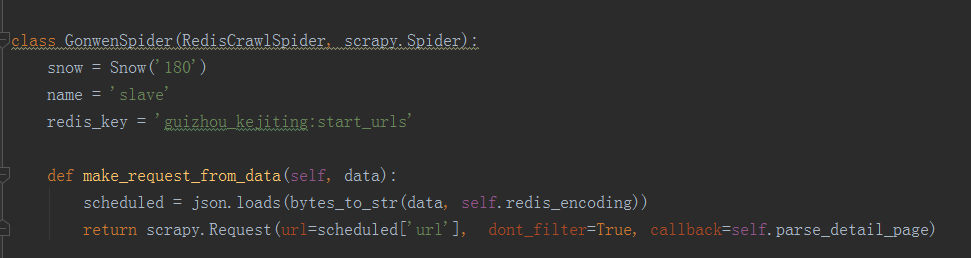
1. 继承的类发生变化，修改为

class GonwenSpider(RedisCrawlSpider, scrapy.Spider):

1. 实例化雪花算法、指定redis数据库



1. 重写函数。



make\_request\_from\_data依据爬虫参数的不同需要做不同的修改.

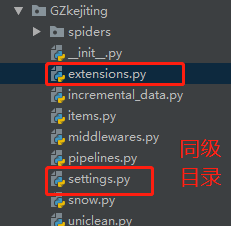
# 关于如何解决从爬虫完成爬虫任务后空跑问题

为了防止在主爬虫没有数据灌入redis数据库时，从爬虫仍然处于监听状态，导致服务器的压力过大，现做如下修改解决从爬虫空跑问题。

**该问题主要是针对从爬虫，只需要按照以下方法修改从爬虫即可。**

## 添加extensions.py文件

注意该文件一定放在与从爬虫的settings同级的目录下。



extensions.py的内容只需要复制分享的extensions.py文件内容即可。或者将分享的extensions.py文件直接copy到上述所说的目录下。

## Settings配置

在从爬虫的settings中添加如下配置（**为了方便，建议先修改工程名**）

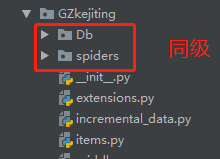


# 关于如何实现本次爬虫爬取数据统计

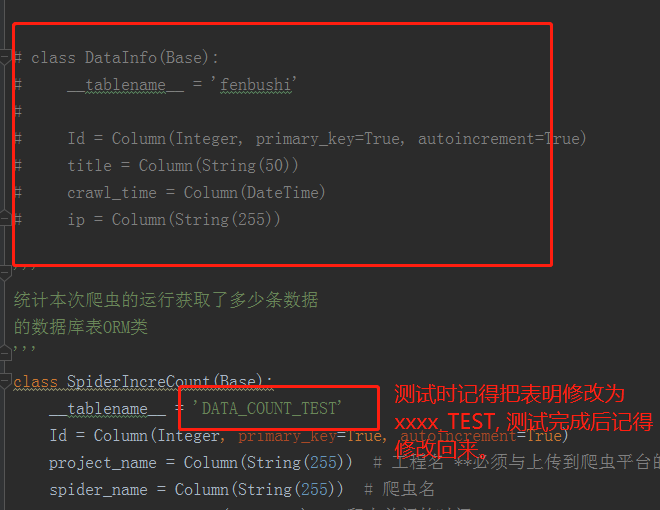
统计目前爬虫平台现有爬取数据情况以及数据存储位置，是为了未来有数据需求、查找数据存放位置等要求时，能够快速获取数据爬取的大致情况。

## 首先建立Db文件用于创建统计数据的数据库表

1. 将分享的Db文件夹copy到与spiders同级的目录下



1. 由于目前用的是mongodb数据库，因此需要将Db文件夹下的DbInfo.py中用于创建存取爬取数据（不是统计爬虫数据的数据库表）的数据库表代码删掉（或注释）。



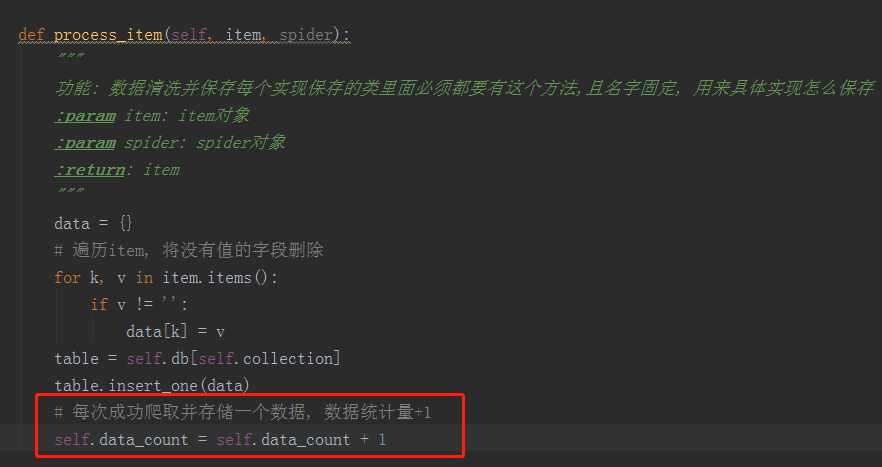
## 写入统计的爬取数据，修改从爬虫的pipelines.py

基于自己的pipelines（**不是分享的pipelines**）作如下修改

1. 修改\_\_init\_\_函数（mongodb的连接参数可以自行设定）



1. 定义gen\_timestamp函数，直接copy分享的代码
2. 修改process\_item函数



1. 修改close\_spider函数



# 统一命名规范（示例）

