**解决 Scrapy-Redis 空跑问题，链接跑完后自动关闭爬虫**

问题：scrapy-redis框架中，reids存储的xxx:requests已经爬取完毕，但程序仍然一直运行，如何自动停止程序，结束空跑。

首先，scrapy内部的信号系统会在爬爬虫的信号管理器收到spider\_idle信号后，将调用注册spider\_idle信号的处理器进行处理。虫耗尽内部队列中的request时，就会触发spider\_idle信号。

其次，爬虫的信号管理器收到spider\_idle信号后，将调用注册spider\_idle信号的处理器进行处理。

再次，当该信号的所有处理器(handler)被调用后，如果spider仍然保持空闲状态，引擎将会关闭该spider；

**解决思路：**

通过前面的了解，我们知道 爬虫关闭的关键是 spider\_idle 信号。

spider\_idle信号只有在爬虫队列为空时才会被触发， 触发间隔为5s。

那么我们也可以使用同样的方式，在信号管理器上注册一个对应在spider\_idle信号下的spider\_idle()方法。

在 spider\_idle() 方法中，编写结束条件来结束爬虫。

**解决方案：**

redis\_key 为空后一段时间关闭爬虫

redis\_key 为空后一段时间关闭爬虫 的实现方案：

这里在 Scrapy 中的 exensions（扩展） 中实现，当然你也可以在pipelines（管道）中实现。

扩展框架提供一个机制，使得你能将自定义功能绑定到Scrapy。 扩展只是正常的类，它们在Scrapy启动时被实例化、初始化。 关于扩展详细见： scrapy 扩展(Extensions)

在settings.py 文件的同级目录下，创建一个名为 extensions.py 的文件，

在其中写入以下代码：

# -\*- coding: utf-8 -\*-

# Define here the models for your scraped Extensions

import logging

import time

from scrapy import signals

from scrapy.exceptions import NotConfigured

logger = logging.getLogger(\_\_name\_\_)

class RedisSpiderSmartIdleClosedExensions(object):

def \_\_init\_\_(self, idle\_number, crawler):

self.crawler = crawler

self.idle\_number = idle\_number

self.idle\_list = []

self.idle\_count = 0

@classmethod

def from\_crawler(cls, crawler):

# 首先检查是否应该启用和提高扩展

# 否则不配置

if not crawler.settings.getbool(‘MYEXT\_ENABLED‘):

raise NotConfigured

# 获取配置中的时间片个数，默认为360个，30分钟

idle\_number = crawler.settings.getint(‘IDLE\_NUMBER‘, 360)

# 实例化扩展对象

ext = cls(idle\_number, crawler)

# 将扩展对象连接到信号， 将signals.spider\_idle 与 spider\_idle() 方法关联起来。

crawler.signals.connect(ext.spider\_opened, signal=signals.spider\_opened)

crawler.signals.connect(ext.spider\_closed, signal=signals.spider\_closed)

crawler.signals.connect(ext.spider\_idle, signal=signals.spider\_idle)

# return the extension object

return ext

def spider\_opened(self, spider):

logger.info("opened spider %s redis spider Idle, Continuous idle limit： %d", spider.name, self.idle\_number)

def spider\_closed(self, spider):

logger.info("closed spider %s, idle count %d , Continuous idle count %d",

spider.name, self.idle\_count, len(self.idle\_list))

def spider\_idle(self, spider):

self.idle\_count += 1 # 空闲计数

self.idle\_list.append(time.time()) # 每次触发 spider\_idle时，记录下触发时间戳

idle\_list\_len = len(self.idle\_list) # 获取当前已经连续触发的次数

# 判断 当前触发时间与上次触发时间 之间的间隔是否大于5秒，如果大于5秒，说明redis 中还有key

if idle\_list\_len > 2 and self.idle\_list[-1] - self.idle\_list[-2] > 6:

self.idle\_list = [self.idle\_list[-1]]

elif idle\_list\_len > self.idle\_number:

# 连续触发的次数达到配置次数后关闭爬虫

logger.info(‘\n continued idle number exceed {} Times‘

‘\n meet the idle shutdown conditions, will close the reptile operation‘

‘\n idle start time: {}, close spider time: {}‘.format(self.idle\_number,

self.idle\_list[0], self.idle\_list[0]))

# 执行关闭爬虫操作

self.crawler.engine.close\_spider(spider, ‘closespider\_pagecount‘)

**在settings.py 中添加以下配置：**

MYEXT\_ENABLED=True # 开启扩展

**# 配置空闲持续时间单位为 360个，一个时间单位为5s，等待半小时没有url，则关闭（依据情况自行设置）**

IDLE\_NUMBER=360

# 在 EXTENSIONS 配置，激活扩展

‘EXTENSIONS‘= {

‘**\*\*\*\*\*\*\*（你的工程名）**.extensions.RedisSpiderSmartIdleClosedExensions‘: 500,

}