#启用Redis调度存储请求队列

SCHEDULER = "scrapy\_redis.scheduler.Scheduler"

#确保所有的爬虫通过Redis去重

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

#默认请求序列化使用的是pickle 但是我们可以更改为其他类似的。PS：这玩意儿2.X的可以用。3.X的不能用

#SCHEDULER\_SERIALIZER = "scrapy\_redis.picklecompat"

#不清除Redis队列、这样可以暂停/恢复 爬取

#SCHEDULER\_PERSIST = True

#使用优先级调度请求队列 （默认使用）

#SCHEDULER\_QUEUE\_CLASS = 'scrapy\_redis.queue.PriorityQueue'

#可选用的其它队列

#SCHEDULER\_QUEUE\_CLASS = 'scrapy\_redis.queue.FifoQueue'

#SCHEDULER\_QUEUE\_CLASS = 'scrapy\_redis.queue.LifoQueue'

#最大空闲时间防止分布式爬虫因为等待而关闭

#这只有当上面设置的队列类是SpiderQueue或SpiderStack时才有效

#并且当您的蜘蛛首次启动时，也可能会阻止同一时间启动（由于队列为空）

#SCHEDULER\_IDLE\_BEFORE\_CLOSE = 10

#将清除的项目在redis进行处理

ITEM\_PIPELINES = {

'scrapy\_redis.pipelines.RedisPipeline': 300

}

#序列化项目管道作为redis Key存储

#REDIS\_ITEMS\_KEY = '%(spider)s:items'

#默认使用ScrapyJSONEncoder进行项目序列化

#You can use any importable path to a callable object.

#REDIS\_ITEMS\_SERIALIZER = 'json.dumps'

#指定连接到redis时使用的端口和地址（可选）

#REDIS\_HOST = 'localhost'

#REDIS\_PORT = 6379

#指定用于连接redis的URL（可选）

#如果设置此项，则此项优先级高于设置的REDIS\_HOST 和 REDIS\_PORT

#REDIS\_URL = 'redis://user:pass@hostname:9001'

#自定义的redis参数（连接超时之类的）

#REDIS\_PARAMS = {}

#自定义redis客户端类

#REDIS\_PARAMS['redis\_cls'] = 'myproject.RedisClient'

#如果为True，则使用redis的'spop'进行操作。

#如果需要避免起始网址列表出现重复，这个选项非常有用。开启此选项urls必须通过sadd添加，否则会出现类型错误。

#REDIS\_START\_URLS\_AS\_SET = False

#RedisSpider和RedisCrawlSpider默认 start\_usls 键

#REDIS\_START\_URLS\_KEY = '%(name)s:start\_urls'

#设置redis使用utf-8之外的编码

#REDIS\_ENCODING = 'latin1'