Scrapy爬虫笔记

前言：为什么爬虫用scrapy框架？

Scrapy框架基于twisted（异步io），性能是最大的优势，scrapy方便扩展，提供了很多内置功能。Scrapy内置的css和xpath selector非常方便，beautifulsoup很慢~。等~~

1. 爬虫的创建

Scrapy框架的下载命令：

pip install Scrapy

创建爬虫的命令

scrapy startproject 爬虫名称

执行爬虫的命令

scrapy crawl 爬虫名称

1. 爬虫的反爬配置

custom\_settings = {   # 自定义每个爬虫的配置文件

        # 不遵守网站规定的爬虫协议

        'ROBOTSTXT\_OBEY': False,

        # 并发请求数量

        'CONCURRENT\_REQUESTS': 16,

        # 请求间的间隔

        'DOWNLOAD\_DELAY': .3,

        # 是否允许Cookie记录相关信息

        'COOKIES\_ENABLED': False,

        # 自定义请求头

        'DEFAULT\_REQUEST\_HEADERS': {

            # 'User-Agent': 'xxxx',

            # 'Cookie': 'xxxx'

        },

        # 缓存爬取的网页（说明：项目完成后取消缓存）

        # 'HTTPCACHE\_ENABLED': True,

        # 'HTTPCACHE\_EXPIRATION\_SECS': 0,

        # 'HTTPCACHE\_DIR':'httpcache',

        # 'HTTPCACHE\_IGNORE\_HTTP\_CODES':[],

        # 'HTTPCACHE\_STORAGE':'scrapy.extensions.httpcache.FilesystemCacheStorage',

    }

1. 爬虫的数据处理

Xpath数据提取的格式

response.xpath("提取的数据位置").extract\_first()

extract和extract\_frist的区别：

同样是解析数据

extract():这个方法返回的是一个数组list，，里面包含了多个string，如果只有一个string，则返回['ABC']这样的形式。可以直接提取列表里的置顶数据

extract\_first()：这个方法返回的是一个string字符串，是list数组里面的第一个字符串。

正则的格式

 pattern = re.compile("decontent\('(.\*?)'")

Scrapy使用yield进行数据传递，如下图

def parse(self, response):

        yield scrapy.Request(

            url=response.url,

            callback=self.parse\_all\_page,

            meta={},

            dont\_filter=True

        )

Callback里写下一个接受数据的函数，meta里写传递的数据

四、管道

1.在settings.py中配置数据库的连接信息，包括MYSQL\_HOST、MYSQL\_DBNAME、MYSQL\_USER、MYSQL\_PASSWORD

2. 手动创建数据库dj72，创建表dj72，包括 id 主键 自增、music\_title、cate\_title、download\_url 唯一

3. 人工修改MySQLPipeline，主要修改sql语句和相关对应的数据

4. 让数据库管道生效，修改settings.py中的ITEM\_PIPELINES ITEM\_PIPELINES = {

   'djyeSpider.pipelines.DownloadPipeline': 300,

   'djyeSpider.pipelines.MySQLPipeline': 350,

}