

Scrapy爬虫大作业报告

1. 实习背景

2022春季，我学习了信息技术基础认知课程python方向的内容。这期间学习了python的基本语法、数据分析库如 numpy、matplotlib、pandas 等库，实现了用beautifulsoup、request等库实现简单的网页爬取，并用 jieba、wordcloud进行词云图的制作。

最后这个大作业采用了scrapy框架进行爬虫的抓取。本报告对于这个项目进行详细说明：

2. 需求分析

1. 爬取豆瓣电影top250的250个电影名称及其信息
2. 使用scrapy框架爬取
3. 使用正则表达式筛选有效信息
4. 输出250个电影的相关信息到csv文件中
5. 作业具体要求
 - 1.正确创建scrapy项目（10分）
 - 2.正确定义scrapys数据模型类（10分）
 - 3.正确定义爬取数据爬虫类（10分）
 - 4.正确定义要爬取的网页xpath表达式爬取数据（20分）
 - 5.将爬取的数据输出到控制台（10分）
 - 6.将爬取的数据解析并存储到json文件（20分）
 - 7.将爬取的数据解析并存储到excel文件（20分）

3. 项目设计

主要参考：

- 课堂讲授
- 课后提供的模板代码
- Scrapy官网<https://www.osgeo.cn/scrapy/topics/items.html>
- 菜鸟教程<https://runoob.com/w3cnote/scrapy-detail.html>
- 实战博客<https://cloud.tencent.com/developer/article/1699680>

3.1 爬取目标

要想爬取一个网页，我们需要对要爬取的网页内容进行分析，便于后续用代码进行处理。

我们的目标URL是：<https://movie.douban.com/top250?start=0&filter=>

打开之后，可以看到每一页有25条电影信息，总共10页，所以需要翻页。检查网页可以发现，每条电影的详细信息在 ol class="grid_view" 下的 li 标签里。

在写scrapy爬虫时，需要构造出10页的URL，生成10次请求，爬取相关内容即可。

3.2 制作爬虫

Scrapy爬虫项目主要有以下几步：

A. 创建项目

```
scrapy startproject mySpider
#事实上对于豆瓣爬取，我起名为：
scrapy startproject Douban_movie_top250
```

mySpider 为项目名称，可以看到将会创建一个 mySpider 文件夹，目录结构大致如下：

各个主要文件的作用：

```
Douban_movie_top250/
  scrapy.cfg
  Douban_movie_top250/
    __init__.py
    items.py
    pipelines.py
    settings.py
    spiders/
      __init__.py
      ...
```

这些文件分别是：

- scrapy.cfg: 项目的配置文件。
- Douban_movie_top250/: 项目的Python模块，将会从这里引用代码。
- Douban_movie_top250/items.py: 项目的目标文件。
- Douban_movie_top250/pipelines.py: 项目的管道文件。
- Douban_movie_top250/settings.py: 项目的设置文件。
- Douban_movie_top250/spiders/: 存储爬虫代码目录。

B. 生成爬虫

```
scrapy genspider Douban movie.douban.com
```

在Spiders文件夹下会看到Douban.py。

根据 **A. 爬取目标** 中的分析，Douban.py需要指明爬取对象：

```
def start_requests(self):
    for i in range(10):
        url = f'https://movie.douban.com/top250?start={25 * i}&filter='
        yield Request(url=url, callback=self.parse)
```

C. 配置爬虫

① 编写items.py

```
import scrapy

class DoubanMovieTop250Item(scrapy.Item):
    name = scrapy.Field()
    pic_link = scrapy.Field()
    rank = scrapy.Field()
    director_actor = scrapy.Field()
    info = scrapy.Field()
    rating_score = scrapy.Field()
    rating_num = scrapy.Field()
    introduce = scrapy.Field()
```

② 编写Douban.py

```
import scrapy
from scrapy import Request
from Douban_movie_top250.items import DoubanMovieTop250Item

class DoubanSpider(scrapy.Spider):
    name = 'Douban'
    allowed_domains = ['movie.douban.com']

    def start_requests(self):
        for i in range(10):
            url = f'https://movie.douban.com/top250?start={25 * i}&filter='
            yield Request(url=url, callback=self.parse)

    def parse(self, response, **kwargs):
        for li in response.xpath("//ol[@class='grid_view']/li"):
            item = DoubanMovieTop250Item()
            item['rank'] =
li.xpath("//div[@class='pic']/em/text()").extract_first()
            item['name'] =
li.xpath("//div[@class='hd']/a/span[@class='title']/text()").extract_first()
            item['pic_link'] =
li.xpath("//div[@class='pic']/a/img/@src").extract_first()
            item['info'] = li.xpath("//div[@class='bd']/p/text()").extract()
[1].strip()
            item['director_actor'] =
li.xpath("//div[@class='bd']/p/text()").extract_first().strip()
            item['rating_score'] =
li.xpath("//div[@class='star']/span[2]/text()").extract_first()
            item['rating_num'] =
li.xpath("//div[@class='star']/span[4]/text()").extract_first()
            item['introduce'] =
li.xpath("//p[@class='quote']/span/text()").extract_first()
            yield item
```

用于指定爬取的url和参数赋值，items.py中设置的参数在这里得到赋值，即我们要爬取的信息种类。

③ 编写pipelines.py

```
from scrapy.pipelines.images import ImagesPipeline # scrapy图片下载器
from scrapy import Request
from scrapy.exceptions import DropItem

class DoubanMovieTop250Pipeline(ImagesPipeline):
    # 请求下载图片
    def get_media_requests(self, item, info):
        yield Request(item['pic_link'], meta={'name': item['name']})

    def item_completed(self, results, item, info):
        # 分析下载结果并剔除下载失败的图片
        image_paths = [x['path'] for ok, x in results if ok]
        if not image_paths:
            raise DropItem("Item contains no images")
        return item

    # 重写file_path方法，将图片以原来的名称和格式进行保存
    def file_path(self, request, response=None, info=None):
        name = request.meta['name'] # 接收上面meta传递过来的图片名称
        file_name = name + '.jpg' # 添加图片后缀名
        return file_name
```

根据博客教程，可以将电影的图片也爬取下来，可以使用pipelines.py。Scrapy提供了专门用于下载的pipeline，支持文件下载和图片下载（异步、多线程）。

④ 配置settings.py

```
# settings.py

BOT_NAME = 'Douban_movie_top250'

SPIDER_MODULES = ['Douban_movie_top250.spiders']
NEWSPIDER_MODULE = 'Douban_movie_top250.spiders'

USER_AGENT = 'Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/535.24 (KHTML, like Gecko) Chrome/19.0.1055.1 Safari/535.24'

ROBOTSTXT_OBEY = False

CONCURRENT_REQUESTS = 10

DOWNLOAD_DELAY = 0.25

ITEM_PIPELINES = {
    'Douban_movie_top250.pipelines.DoubanMovieTop250Pipeline': 300,
}

IMAGES_STORE = './Douban_pic'
```

设置一些爬取的参数，如下载频率、请求次数等。

D. 运行爬虫

```
# 终端命令
scrapy crawl Douban -o movies_info.csv
```

接着pycharm的终端中就开始输出一些爬虫属性，接着停顿之后就会进行爬取，爬取完成后会看到Douban_pic文件夹和movies_info.csv文件。

这个过程需要把梯子关掉，不然会报错。

5. 项目编码

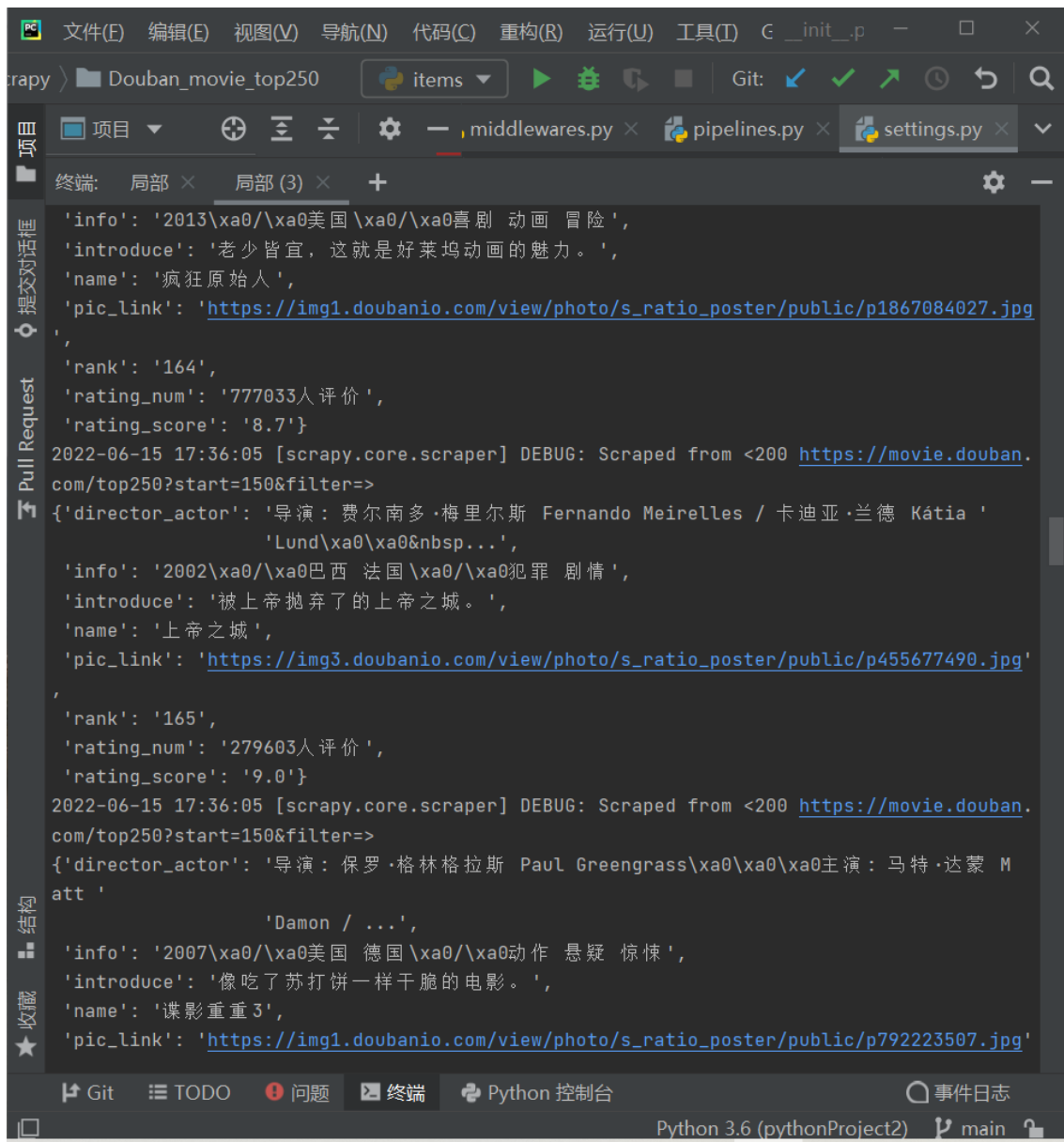
- 编程工具：Pycharm 社区版 2020.2
- 语言：Python
- 库：scrapy

6. 项目测试

开着梯子是无法正常爬取的，豆瓣应该会屏蔽掉国外的请求，关掉之后不需要重启程序即能继续爬取。

项目运行结果如下：

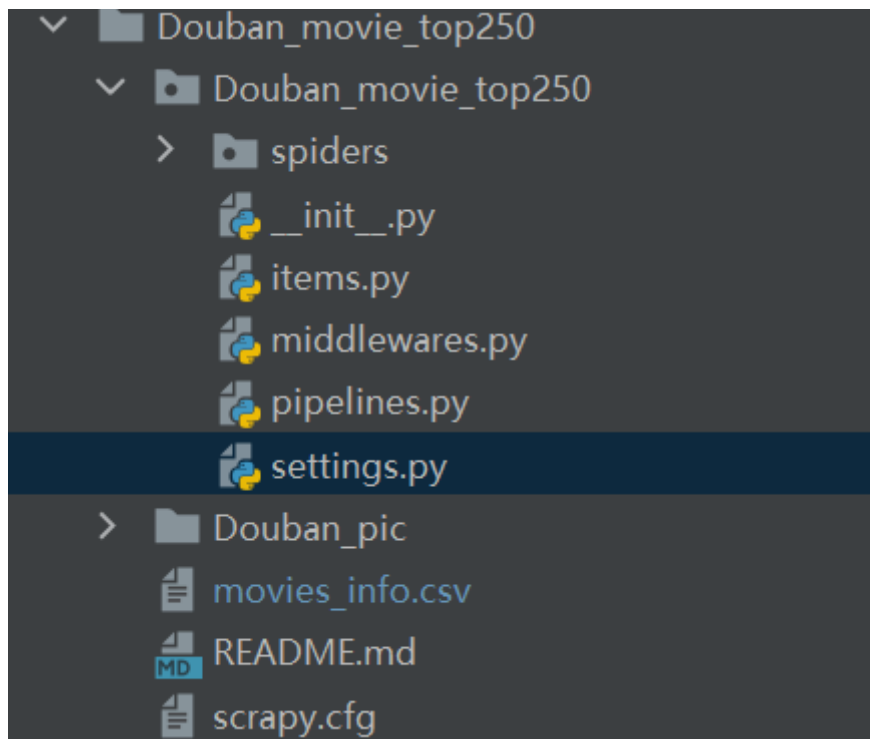
- 运行过程



The screenshot shows the PyCharm IDE interface. The terminal window at the bottom displays the following output:

```
2022-06-15 17:36:05 [scrapy.core.scraper] DEBUG: Scraped from <200 https://movie.douban.com/top250?start=150&filter=>
{'director_actor': '导演：费尔南多·梅里尔斯 Fernando Meirelles / 卡迪亚·兰德 Kátia '
                  'Lund\\xa0\\xa0&nbsp...',
 'info': '2002\\xa0\\xa0巴西 法国\\xa0\\xa0犯罪 剧情',
 'introduce': '被上帝抛弃了的上帝之城。',
 'name': '上帝之城',
 'pic_link': 'https://img3.doubanio.com/view/photo/s_ratio_poster/public/p455677490.jpg',
 'rank': '164',
 'rating_num': '777033人评价',
 'rating_score': '8.7'}
2022-06-15 17:36:05 [scrapy.core.scraper] DEBUG: Scraped from <200 https://movie.douban.com/top250?start=150&filter=>
{'director_actor': '导演：保罗·格林格拉斯 Paul Greengrass\\xa0\\xa0\\xa0主演：马特·达蒙 M
att '
                  'Damon / ...',
 'info': '2007\\xa0\\xa0美国 德国\\xa0\\xa0动作 悬疑 惊悚',
 'introduce': '像吃了苏打饼一样干脆的电影。',
 'name': '谍影重重3',
 'pic_link': 'https://img1.doubanio.com/view/photo/s_ratio_poster/public/p792223507.jpg'}
```

- 运行结果目录



- 后续处理

由于直接生成了 csv , 为了完成老师的需求, 把 csv 转换成 json 吧...

编写 csv_json1.py, 在运行整个项目后单独执行来进行转换。

```
import pandas as pd
path =
"D:\\Users\\shandaiwang\\PycharmProjects\\pythonProject2\\douban_top250_scrapy\\douban_top250_scrapy\\Douban_movie_top250\\movies_info.csv"
# path1 = "Douban_movie_top250\\movies_info.csv"
csv_data = pd.read_csv(path)
csv_data.to_json("movies_info.json", orient = "records")
```

这里使用的是绝对路径来读取, 异地运行需要调整 path 的值。