

Day06回顾

京东爬虫

- 执行JS脚本,把进度条拉到最下面

```
1 1、js脚本
2 browser.execute_script(
3     'window.scrollTo(0,document.body.scrollHeight)'
4 )
5 2、利用节点对象的text属性获取当前节点及后代节点的文本内容,想办法处理数据
```

- 设置无界面模式 (chromedriver | firefox)

```
1 options = webdriver.ChromeOptions()
2 options.add_argument('--headless')
3
4 browser = webdriver.Chrome(options=options)
5 browser.get(url)
```

多线程爬虫

- 使用流程

```
1 # 1、URL队列
2 q.put(url)
3 # 2、线程事件函数
4 while True:
5     if not url_queue.empty():
6         ...get()、请求、解析
7     else:
8         break
9 # 创建并启动线程
10 t_list = []
11 for i in range(5):
12     t = Thread(target=parse_page)
13     t_list.append(t)
14     t.start()
15 # 阻塞等待回收线程
16 for i in t_list:
```

json模块

▪ json转python

```
1 变量名 = json.loads(res.text))
```

▪ python转json (保存为json文件)

```
1  # 保存所抓取数据为json数据
2  with open(filename, 'a') as f:
3      json.dump(字典/列表/元组, f, ensure_ascii=False)
```

scrapy框架

▪ 五大组件

```
1 引擎 (Engine)
2 爬虫程序 (Spider)
3 调度器 (Scheduler)
4 下载器 (Downloader)
5 管道文件 (Pipeline)
6  # 两个中间件
7  下载器中间件 (Downloader Middlewares)
8  蜘蛛中间件 (Spider Middlewares)
```

▪ 工作流程

```
1  1、Engine向Spider索要URL, 交给Scheduler入队列
2  2、Scheduler处理后出队列, 通过Downloader Middlewares交给Downloader去下载
3  3、Downloader得到响应后, 通过Spider Middlewares交给Spider
4  4、Spider数据提取:
5      1、数据交给Pipeline处理
6      2、需要跟进URL, 继续交给Scheduler入队列, 依次循环
```

▪ 常用命令

```
1 # 创建爬虫项目
2 scrapy startproject 项目名
3
4 # 创建爬虫文件
5 cd 项目文件夹
6 scrapy genspider 爬虫名 域名
7
8 # 运行爬虫
9 scrapy crawl 爬虫名
```

■ scrapy项目目录结构

```
1 Baidu
2 |— Baidu          # 项目目录
3 |   |— items.py    # 定义数据结构
4 |   |— middlewares.py # 中间件
5 |   |— pipelines.py # 数据处理
6 |   |— settings.py  # 全局配置
7 |   |— spiders
8 |       |— baidu.py # 爬虫文件
9 |— scrapy.cfg      # 项目基本配置文件
```

■ settings.py全局配置

```
1 1、USER_AGENT = 'Mozilla/5.0'
2 2、ROBOTSTXT_OBEY = False
3 3、CONCURRENT_REQUESTS = 32
4 4、DOWNLOAD_DELAY = 1
5 5、DEFAULT_REQUEST_HEADERS={}
6 6、ITEM_PIPELINES={'项目目录名.pipelines.类名':300}
```

Day07笔记

scrapy框架

■ 创建爬虫项目步骤

```
1 1、新建项目：scrapy startproject 项目名
2 2、cd 项目文件夹
3 3、新建爬虫文件：scrapy genspider 文件名 域名
4 4、明确目标(items.py)
5 5、写爬虫程序(文件名.py)
6 6、管道文件(pipelines.py)
7 7、全局配置(settings.py)
8 8、运行爬虫：scrapy crawl 爬虫名
```

- pycharm运行爬虫项目

```
1 1、创建begin.py(和scrapy.cfg文件同目录)
2 2、begin.py中内容:
3     from scrapy import cmdline
4     cmdline.execute('scrapy crawl maoyan'.split())
```

小试牛刀

- 目标

```
1 打开百度首页, 把 '百度一下, 你就知道' 抓取下来, 从终端输出
```

- 实现步骤

1. 创建项目Baidu 和 爬虫文件baidu

```
1 1、 scrapy startproject Baidu
2 2、 cd Baidu
3 3、 scrapy genspider baidu www.baidu.com
```

2. 编写爬虫文件baidu.py, xpath提取数据

```
1 # -*- coding: utf-8 -*-
2 import scrapy
3
4 class BaiduSpider(scrapy.Spider):
5     name = 'baidu'
6     allowed_domains = ['www.baidu.com']
7     start_urls = ['http://www.baidu.com/']
8
9     def parse(self, response):
10         result = response.xpath('/html/head/title/text()').extract_first()
11         print('*'*50)
12         print(result)
13         print('*'*50)
14
```

3. 全局配置settings.py

```
1 USER_AGENT = 'Mozilla/5.0'
2 ROBOTSTXT_OBEY = False
3 DEFAULT_REQUEST_HEADERS = {
4     'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
5     'Accept-Language': 'en',
6 }
```

4. 创建begin.py (和scrapy.cfg同目录)

```
1 from scrapy import cmdline
2
3 cmdline.execute('scrapy crawl baidu'.split())
```

5. 启动爬虫

```
1 直接运行 begin.py 文件即可
```

思考运行过程

猫眼电影案例

■ 目标

```
1 URL: 百度搜索 -> 猫眼电影 -> 榜单 -> top100榜
2 内容: 电影名称、电影主演、上映时间
```

■ 实现步骤

1. 创建项目和爬虫文件

```
1 # 创建爬虫项目
2
3 # 创建爬虫文件
4
```

2. 定义要爬取的数据结构 (items.py)

```
1 |
```

3. 编写爬虫文件 (maoyan.py)

```
1 1、基准xpath, 匹配每个电影信息节点对象列表
2     dd_list = response.xpath('//dl[@class="board-wrapper"]/dd')
3 2、for dd in dd_list:
4     电影名称 = dd.xpath('./a/@title')
5     电影主演 = dd.xpath('./p[@class="star"]/text()')
6     上映时间 = dd.xpath('./p[@class="releasetime"]/text()')
```

代码实现一

```
1 |
```

代码实现二

```
1 |
```

代码实现三

```
1 |
```

3. 定义管道文件 (pipelines.py)

```
1 |
```

5. 全局配置文件 (settings.py)

```
1 |
```

6. 创建并运行文件 (begin.py)

```
1 from scrapy import cmdline
2 cmdline.execute('scrapy crawl maoyan'.split())
```

知识点汇总

■ 节点对象.xpath("")

```
1 1、列表,元素为选择器 ['<selector data='A'>]
2 2、列表.extract() : 序列化列表中所有选择器为Unicode字符串 ['A','B','C']
3 3、列表.extract_first() 或者 get() :获取列表中第1个序列化的元素(字符串)
```

■ pipelines.py中必须由1个函数叫process_item

```
1 def process_item(self,item,spider):
2     return item ( * 此处必须返回 item )
```

■ 日志变量及日志级别(settings.py)

```
1 # 日志相关变量
2 LOG_LEVEL = ''
3 LOG_FILE = '文件名.log'
4
5 # 日志级别
6 5 CRITICAL : 严重错误
7 4 ERROR    : 普通错误
8 3 WARNING  : 警告
9 2 INFO     : 一般信息
10 1 DEBUG    : 调试信息
11 # 注意: 只显示当前级别的日志和比当前级别日志更严重的
```

■ 管道文件使用

```
1 1、在爬虫文件中为items.py中类做实例化，用爬下来的数据给对象赋值
2     from ..items import MaoyanItem
3     item = MaoyanItem()
4 2、管道文件 (pipelines.py)
5 3、开启管道 (settings.py)
6     ITEM_PIPELINES = { '项目目录名.pipelines.类名':优先级 }
```

数据持久化存储(MongoDB和MySQL)

实现步骤

```
1 1、在setting.py中定义相关变量
2 2、pipelines.py中导入settings模块
3     def open_spider(self,spider):
4         # 爬虫开始执行1次,用于数据库连接
5     def close_spider(self,spider):
6         # 爬虫结束时执行1次,用于断开数据库连接
7 3、settings.py中添加此管道
8     ITEM_PIPELINES = {'':200}
9
10 # 注意：process_item() 函数中一定要 return item ***
```

保存为csv、json文件

■ 命令格式

```
1 scrapy crawl maoyan -o maoyan.csv
2 scrapy crawl maoyan -o maoyan.json
```

盗墓笔记小说抓取案例（三级页面）

■ 目标

```
1 # 抓取目标网站中盗墓笔记1-8中所有章节的所有小说的具体内容，保存到本地文件
2 1、网址：http://www.daomubiji.com/
```

■ 准备工作xpath

```
1 1、一级页面xpath (此处响应做了处理) : //ul[@class="sub-menu"]/li/a/@href
2 2、二级页面xpath: /html/body/section/div[2]/div/article
3 基准xpath : //article
4 3、三级页面xpath: response.xpath('//article[@class="article-content"]//p/text()').extract()
```

■ 项目实施

1. 创建项目及爬虫文件

```
1 创建项目 : Daomu
2 创建爬虫 : daomu www.daomubiji.com
```

2. 定义要爬取的数据结构 (把数据交给管道)

```
1 import scrapy
2
3 class DaomuItem(scrapy.Item):
4     # 卷名
5     juan_name = scrapy.Field()
6     # 章节数
7     zh_num = scrapy.Field()
8     # 章节名
9     zh_name = scrapy.Field()
10    # 章节链接
11    zh_link = scrapy.Field()
12    # 小说内容
13    zh_content = scrapy.Field()
```

3. 爬虫文件实现数据抓取

```
1 |
```

4. 管道文件实现数据处理

```
1 |
```

图片管道(360图片抓取案例)

■ 目标

```
1 | www.so.com -> 图片 -> 美女
```

■ 抓取网络数据包


```
1 2、F12抓包,抓取到json地址 和 查询参数(QueryString)
2     url = 'http://image.so.com/zj?ch=beauty&sn={}&listtype=new&temp=1'.format(str(sn))
3     ch: beauty
4     sn: 90
5     listtype: new
6     temp: 1
```

■ 项目实施

1. 创建爬虫项目和爬虫文件

```
1 |
```

2. 定义要爬取的数据结构

```
1 |
```

3. 爬虫文件实现图片链接抓取

```
1 |
```

4. 管道文件 (pipelines.py)

```
1 |
```

5. 设置settings.py

```
1 |
```

6. 创建begin.py运行爬虫

```
1 |
```

今日作业

```
1 1、把今天内容过一遍
2 2、Daomu错误调一下(看规律,做条件判断)
3 3、腾讯招聘尝试改写为scrapy
4     response.text : 获取页面响应内容
```

