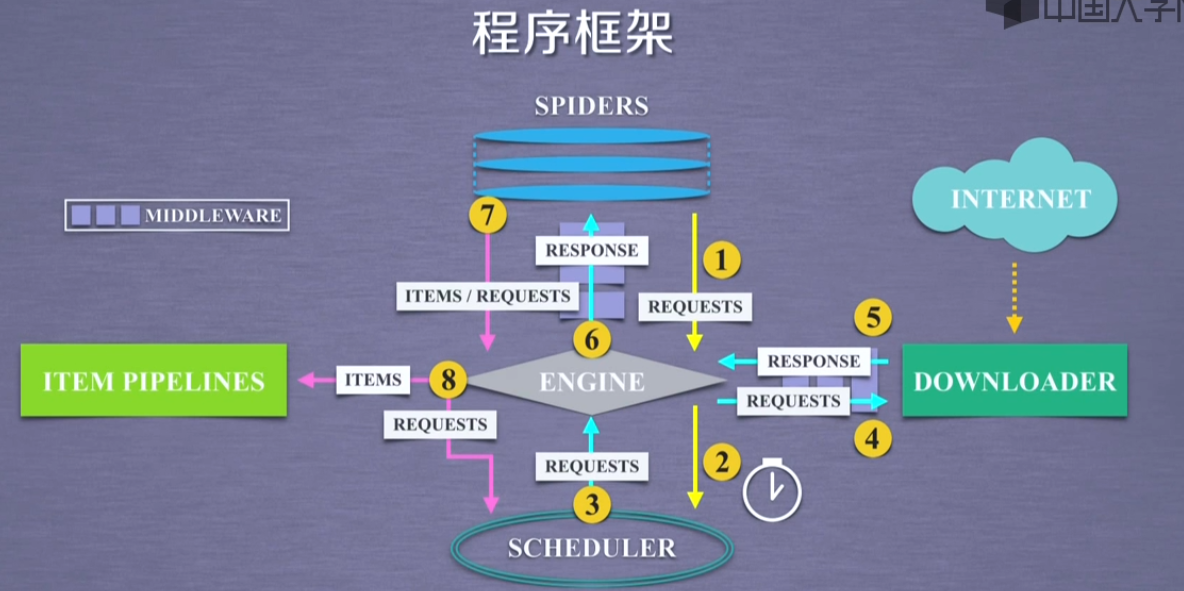
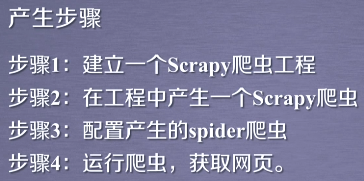
Scrapy爬虫框架实例

# Scrapy框架回顾



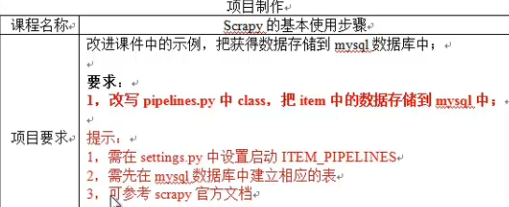
  


# 项目制作

## 第一个



## 第二个

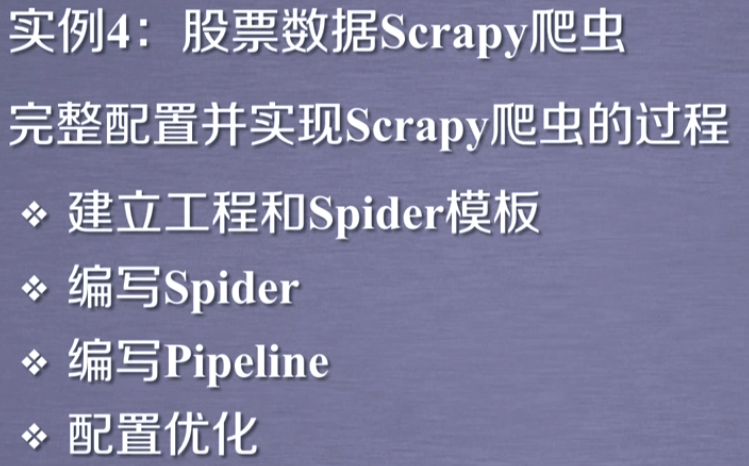


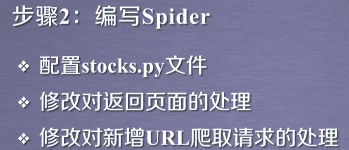
# 股票数据Scrapy爬虫

<https://gupiao.baidu.com/>

<http://finance.sina.com.cn/stock/>

<http://quote.eastmoney.com/stocklist.htm>





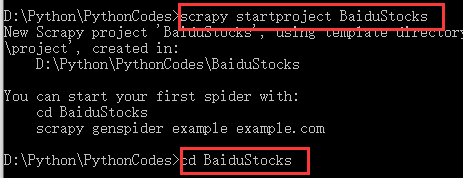
## 步骤1：创建工程

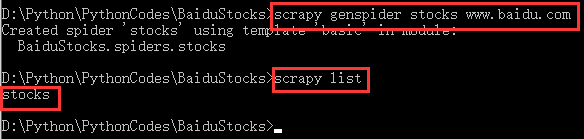
进入到特定目录下：scrapy startproject BaiduStocks

cd BaiduStocks

scrapy genspider stocks baidu.com

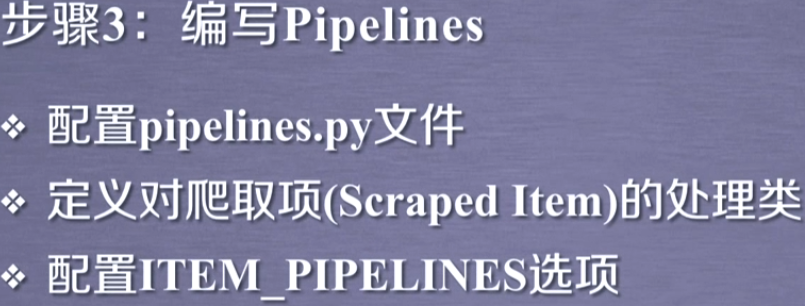
scrapy list 可以查看爬虫工程





## 步骤2：编写stocks.py文件

## 步骤3：编写pipelines.py并配置settings.py文件



参考代码：

stocks.py 文件

*# -\*- coding: utf-8 -\*-***import** scrapy  
**import** re  
  
**class** StocksSpider(scrapy.Spider):  
 name = **"stocks"** start\_urls = [**'http://quote.eastmoney.com/stocklist.html'**]  
  
 **def** parse(self, response):  
 **for** href **in** response.css(**'a::attr(href)'**).extract():  
 **try**:  
 stock = re.findall(**r"[s][hz]\d{6}"**, href)[0]  
 url = **'https://gupiao.baidu.com/stock/'** + stock + **'.html'  
 yield** scrapy.Request(url, callback=self.parse\_stock)  
 **except**:  
 **continue  
  
 def** parse\_stock(self, response):  
 infoDict = {}  
 stockInfo = response.css(**'.stock-bets'**)  
 name = stockInfo.css(**'.bets-name'**).extract()[0]  
 keyList = stockInfo.css(**'dt'**).extract()  
 valueList = stockInfo.css(**'dd'**).extract()  
 **for** i **in** range(len(keyList)):  
 key = re.findall(**r'>.\*</dt>'**, keyList[i])[0][1:-5]  
 **try**:  
 val = re.findall(**r'\d+\.?.\*</dd>'**, valueList[i])[0][0:-5]  
 **except**:  
 val = **'--'** infoDict[key]=val  
  
 infoDict.update(  
 {**'股票名称'**: re.findall(**'\s.\*\('**,name)[0].split()[0] + \  
 re.findall(**'\>.\*\<'**, name)[0][1:-1]})  
 **yield** infoDict

**pipelines.py文件**：

**class** BaidustocksPipeline(object):  
 **def** process\_item(self, item, spider):  
 **return** item  
  
**class** BaidustocksInfoPipeline(object):  
 **def** open\_spider(self, spider):  
 self.f = open(**'BaiduStockInfo.txt'**, **'w'**)  
  
 **def** close\_spider(self, spider):  
 self.f.close()  
  
 **def** process\_item(self, item, spider):  
 **try**:  
 line = str(dict(item)) + **'\n'** self.f.write(line)  
 **except**:  
 **pass  
 return** item

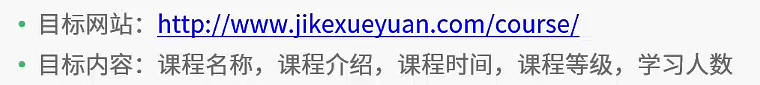
配置settings.py文件，并进行必要的优化



优化略。

# 实例：

爬取极客学院的所有的课程信息。



利用scrapy框架：

## items.py：

**import** scrapy  
**class** JikecrawlItem(scrapy.Item):  
 COURSE\_NAME = scrapy.Field()  
 COURSE\_INTRO = scrapy.Field()  
 COURSE\_TIME = scrapy.Field()  
 COURSE\_RANK = scrapy.Field()  
 COURSE\_NUM = scrapy.Field()

## settings.py

LOG\_FILE = **"./log.txt"**

*# Obey robots.txt rules*ROBOTSTXT\_OBEY = **False**

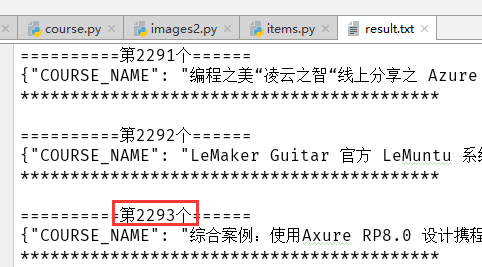
ITEM\_PIPELINES = {  
 **'jikeCrawl.pipelines.JikecrawlPipeline'**: 1,  
}

## pipiline.py

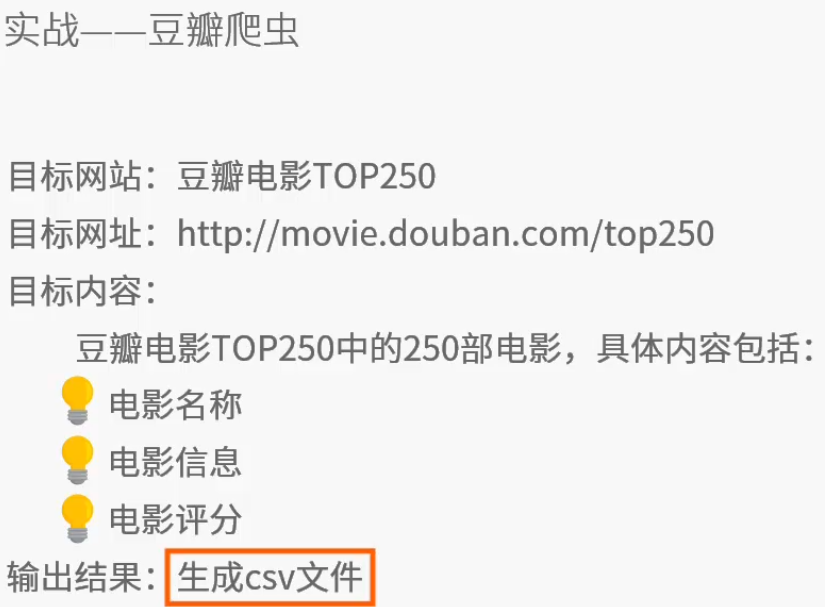
**import** json  
**class** JikecrawlPipeline(object):  
 num = 0  
 **def** process\_item(self, item, spider):  
 self.num += 1  
 itemJson = json.dumps(dict(item), ensure\_ascii=**False**)  
 **with** open(**'./result.txt'**,**'a'**,encoding=**'utf-8'**) **as** f:  
 f.write(**'==========第%d个======'**%self.num+**'\n'**)  
 f.write(itemJson+**'\n'**+**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**+**'\n\n'**)  
 f.close()  
 **return** item

## spider.py 文件

*# -\*- coding: utf-8 -\*-***import** scrapy  
**from** jikeCrawl.items **import** JikecrawlItem  
**import** re  
**import** logging  
  
**class** CourseSpider(scrapy.Spider):  
 name = **'course'  
 def** start\_requests(self):  
 url = **'http://www.jikexueyuan.com/course/'  
 yield** scrapy.Request(url = url,callback=self.parse,meta={**'again'**:**'True'**})  
  
 **def** parse(self, response):  
 item = JikecrawlItem();  
 self.writeToTxt(response.text)  
 **try**:  
 liList = response.xpath(**'//li[@id and @test and @deg]'**)  
 nameList = liList.xpath(**'div[@class="lessonimg-box"]/a/img/@title'**).extract()  
 introList = liList.xpath(**'div[@class="lesson-infor"]/p/text()'**).extract()  
 divList = liList.xpath(**'div[@class="lesson-infor"]/div[@class="timeandicon"]/div[@class="cf"]'**)  
 timeList =divList.xpath(**'dl/dd[@class="mar-b8"]/em/text()'**).extract()  
 rankList = divList.xpath(**'dl/dd[@class="zhongji"]/em/text()'**).extract()  
 numList = divList.xpath(**'em[@class="learn-number"]/text()'**).extract()  
 **for** i **in** range(len(nameList)):  
 logging.info(i)  
 item[**'COURSE\_NAME'**] = nameList[i]  
 item[**'COURSE\_INTRO'**] = introList[i]  
 item[**'COURSE\_RANK'**] = rankList[i]  
 item[**'COURSE\_TIME'**] = timeList[i]  
 item[**'COURSE\_NUM'**] = re.findall(**'\d+'**,numList[i])[0]  
 logging.info(**'%%%%%%%%%%%%%%%%%%%%%%%%%%'**)  
 **yield** item  
 **if** response.meta[**'again'**] == **'True'**:  
 logging.info(**'#############################'**)  
 **for** j **in** range(97):  
 **if** j > 1:  
 **yield** scrapy.Request(url = response.url+**'?pageNum='**+str(j),callback=self.parse,meta={**'again'**:**'False'**})  
 **else**:  
 **continue  
 except** Exception **as** e:  
 logging.error(str(e))  
  
 **def** writeToTxt(self,text):  
 **with** open(**"./html.html"**,**'w'**,encoding=**'utf-8'**) **as** f:  
 f.write(text)



# 豆瓣爬虫



## items.py

import scrapy  
class DoubanmovieItem(scrapy.Item):  
 MOVIE\_NAME = scrapy.Field()  
 MOVIE\_INTRO = scrapy.Field()  
 MOVIE\_SCORE = scrapy.Field()  
 MOVIE\_QUOTE = scrapy.Field()  
 MOVIE\_PEOPLENUM = scrapy.Field()

## pipelines.py

import json  
  
class DoubanmoviePipeline(object):  
 def process\_item(self, item, spider):  
 f = open('./result.txt','a',encoding='utf-8')  
 strJson = json.dumps(dict(item),ensure\_ascii=False)  
 f.write(strJson+'\n')  
 f.close()  
 return item

## settings.py

ROBOTSTXT\_OBEY = False  
LOG\_FILE = './log.txt'

ITEM\_PIPELINES = {  
 'doubanMovie.pipelines.DoubanmoviePipeline': 300,  
}

FEED\_URI = './douban.csv'  
FEED\_FORMAT = 'csv' # 在当前目录下，自动生成csv文件，不影响pipeline的处理

## movies.py

# -\*- coding: utf-8 -\*-  
import scrapy  
import logging  
from doubanMovie.items import DoubanmovieItem  
import re  
  
logger = logging.getLogger('赵红')  
class MoviesSpider(scrapy.Spider):  
 name = 'movies'  
 # allowed\_domains = ['douban.com']  
 def start\_requests(self):  
 url = 'https://movie.douban.com/top250?start=0&filter='  
 header = {'user-agent':'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.221 Safari/537.36 SE 2.X MetaSr 1.0'}  
 num = 0  
 while num < 25:  
 url = re.sub('start=\d+','start='+str(num),url)  
 logger.info("新的URL ："+url)  
 yield scrapy.Request(url = url,callback=self.parse,headers=header)  
 num += 25  
  
 def parse(self, response):  
 logger.info(response.status)  
 f = open('./html.html','w',encoding='utf-8')  
 f.write(response.text)  
 logger.info('Successfully')  
 f.close()  
  
 item = DoubanmovieItem()  
 divList = response.xpath('//ol[@class="grid\_view"]/li/div[@class="item"]')  
 logger.info(str(len(divList))+'AAAAAAAAAAAAA')  
 infoList = divList.xpath('div[@class="info"]')  
 logger.info(str(len(infoList))+'AAAAAAAAAAAAABBB')  
 # nameList = infoList.xpath('div[@class="hd"]/a/span[1]').xpath('string()').extract()  
 # nameList = infoList.xpath('div[@class="hd"]/a/span[1]/text()').extract()  
 nameList = infoList.xpath('div[@class="hd"]/a/span[1]').xpath('text()').extract()  
 logger.info(str(len(nameList))+'YYYYYYYYYYYYYYY')  
 bdList = infoList.xpath('div[@class="bd"]')  
 introList = bdList.xpath('p[@class=""]').xpath('string()').extract()  
 logger.info(str(len(introList))+'CCCCCCCCCCC')  
 scoreList = bdList.xpath('div[@class="star"]/span[@class="rating\_num"]/text()').extract()  
 logger.info(str(len(scoreList))+'IIIIIIIIIIIIIIIIIIIIIIIII')  
 peopleNumList = bdList.xpath('div[@class="star"]/span[last()]/text()').extract()  
 logger.info(str(len(peopleNumList))+'UUUUUUUUUUUUUU')  
 quoteList = bdList.xpath('p[@class="quote"]/span[@class="inq"]/text()').extract()  
 logger.info(str(len(quoteList))+'OOOOOOOOOOOOOOOOOOOO')  
  
 for i in range(len(nameList)):  
 item['MOVIE\_NAME'] = nameList[i].strip(' \n')  
 item['MOVIE\_INTRO'] = introList[i].strip(' \n')  
 item['MOVIE\_SCORE'] = scoreList[i].strip(' \n')  
 item['MOVIE\_QUOTE'] = quoteList[i].strip(' \n')  
 item['MOVIE\_PEOPLENUM'] = re.findall('\d+',peopleNumList[i].strip(' \n'))[0]  
 yield item

## main.py

from scrapy import cmdline  
# cmdline.execute('scrapy list'.split())  
cmdline.execute("scrapy crawl movies".split())

## 结果result.txt

