1. scrapy安装
2. scrapy架构
3. 创建一个项目

scrapy startproject qsbk

初始目录如下

|-qsbk

|-scrapy.cfg

|-qsbk

  |-\_\_init\_\_.py

  |-items.py

  |-middlewares.py

  |-pipelines.py

  |-settings.py

  |-spiders

    |-\_\_init\_\_.py

1. 创建一个爬虫

cd qsbk

scrapy genspider qsbk\_spider "qiushibaike.com"

1. 执行爬虫

在根目录下添加start.py 脚本

内容如下：

from scrapy import cmdline

cmdline.execute(["scrapy","crawl","qsbk\_spider"]);

执行：

python start.py

1. 设置settings.py参数
2. **USER\_AGENT**

USER\_AGENT = ‘Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_13\_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.90 Safari/537.36’

USER\_AGENT的值，可以先请求百度，在请求头中获取。

1. **ROBOTSTXT\_OBEY**

ROBOTSTXT\_OBEY = False

默认为True

False:表示不遵守该网站的robots协议。有一定法律风险。

DOWNLOAD\_DELAY

DOWNLOAD\_DELAY = 1

设置下载延迟，防止爬取的网站负载过高挂了。

1. **ITEM\_PIPELINES**

ITEM\_PIPELINES = {

'qsbk.pipelines.QsbkPipeline': 300,

}

当使用pipelines和items数据模板时，需要打开ITEM\_PIPELINES的设置。

300表示pipelines优先级，可以有多个pipelines，越小表示优先级越高。

1. 主要源代码
2. 根目录/qsbk/pipelines.py

import json  
  
class QsbkPipeline(object):  
 def \_\_init\_\_(self):  
 self.fp = open("duanzi.json",'w',encoding='utf-8')  
 def open\_spider(self,spider):  
 print("爬虫开始。。。")  
 def process\_item(self, item, spider):  
 item\_json = json.dumps(dict(item),ensure\_ascii=False)  
 self.fp.write(item\_json + "\n");  
 return item  
 def close\_spider(self,spider):  
 self.fp.close()  
 print("爬虫结束。。。")

1. 根目录/qsbk/items.py

import scrapy  
  
  
class QsbkItem(scrapy.Item):  
 # define the fields for your item here like:  
 # name = scrapy.Field()  
 author = scrapy.Field()  
 content = scrapy.Field()

1. 根目录/qsbk/spiders/

import scrapy  
from scrapy.http.response.html import HtmlResponse  
from scrapy.selector.unified import SelectorList  
from qsbk.items import QsbkItem  
  
class QsbkSpiderSpider(scrapy.Spider):  
 name = 'qsbk\_spider'  
 allowed\_domains = ['qiushibaike.com']  
 start\_urls = ['https://www.qiushibaike.com/text/page/1']  
 base\_domain = "https://www.qiushibaike.com"  
  
 def parse(self, response):  
 datas = response.xpath("//div[@class='col1 old-style-col1']/div")  
 for data in datas:  
 author = data.xpath(".//div[@class='author clearfix']//h2/text()").get().strip()  
 content = data.xpath(".//a[@class='contentHerf']/div[@class='content']//text()").getall()  
 content = "".join(content).strip()  
 item = QsbkItem(author=author,content=content)  
 yield item  
 next\_url = response.xpath("//ul[@class='pagination']/li[last()]/a/@href").get()  
 print("=======next\_url======")  
 print(next\_url)  
 if not next\_url:  
 return  
 else:  
 yield scrapy.Request(self.base\_domain + next\_url,callback=self.parse)