



目录：

- 1.数据写入csv文件
- 2.数据写入excel文件
- 3.数据存入mongodb
- 4.数据写入mysql
- 5.影评爬取代码展示

注意：爬取的数据以字典的形式放入了self.data属性中[{}, {}, {}]

如需直接使用需要把数据类型转换为列表，把self.data换成自己的数据存储变量名

1.数据写入csv文件

爬取的数据以字典的形式放入了self.data属性中

```
1 #获取header为第一行的字段名
2 header = self.data[0].keys()
3 # 写入文件test.csv,
4 with open('test.csv', 'w', encoding='utf8') as f:
5     # 提前预览列名，当下面代码写入数据时，会将其一一对应。
6     witer = csv.DictWriter(f, fieldnames=header)
7     # 写入列名
8     witer.writeheader()
9     # 写入字典中的数据
10    witer.writerows(self.data)
```

2.数据写入excel文件

爬取的数据以字典的形式放入了self.data属性中

需要应用xlwt的包

```

2 #实例化一个xlwt对象
workbook = xlwt.Workbook(encoding='utf8')
3 # 新建一个excel书签, 名为test
4 sheet1 = workbook.add_sheet('test')
5 # 获取字段
header = list(self.data[0].keys())
6 # 使用for循环, 字段数的长度和字段
7 for i, key in zip(range(len(header)), header):
8 # excel写入字段名
9     sheet1.write(0, i, key)
10
11 #使用两个for循环, j用来控制写入数据的excel的行数, m用来控制列数, 而key则是用来取字典的数据
12 for j in range(1, len(self.data) + 1):
13     for m, key in zip(range(len(header)), header):
14         #写入数据
15         sheet1.write(j, m, self.data[j - 1][key])
16
17 #保存文件路径, 不用新建excel文件, 自动生成
18 workbook.save(r'C:\Users\Administrator\Desktop\test.xlsx')
19
20

```



3.数据存入mongodb

爬取的数据还是存放在类的self.data属性中

mongodb数据库没有表的概念, 数据的写入相对较为简单一点

```

1
2 #连接mongodb
3 mdb = MongoClient('127.0.0.1', 27017)
4
5 # 指定mongodb的库的集合名称, 不用新建, 运行后直接创建
6 db = mdb.test['moive_info']
7
8 #for循环 (self.data的数据类型为列表, 中间的数据格式为字典[{}, {}]) 拿取数据字典格式
9 for i in self.data:
10
11 #判断数据是否为字典类型, 如果是, 写入
12     if isinstance(i, dict):
13         db.insert_one(i)
14
15     #如果不是的, 则报i的title字段+写入数据失败
16     else:
17         print(i['title'], '写入失败')
18
19
20

```



4.数据写入mysql

爬取的数据还是存放在类的self.data属性中

```

1
2 # 连接mysql数据库, 注: 注意指定database的名称
3 db = pymysql.connect(host='127.0.0.1', user='root', password='123456', port=3306, database='test')
4 #实例化一个游标对象
5 cursor = db.cursor()
6 #创建两个字段, keys为sql语句的字段名, values为数据的字符串

```

```

5 keys = ','.join(self.data[0].keys())
6 values = ','.join(['%s' * len(self.data[0])])
7 #这里的mysql数据库的表需要提前创建好
7 # 利用上面的站位字符串, 构造一个通用sql语句
8 sql = 'insert into moive_list(id,{key}) values(null,{values})'.format(key=keys, values=values)
9 # for循环把数据取成字典类型
10 for i in self.data:
11     # 先获取字典值的数据为value, 每一个数据赋值给一个变量
12     value = list(i.values())
13     title = value[0][0]
14     start = value[1][0]
15     actor = value[2]
16     # 写一个报错, 增加容错率
17     try:
18         if cursor.execute(sql, (title, start, actor)):
19             print(title, 'seccess')
20         except:
21             db.rollback()
22
23 #需要使用这个db.commit(), sql语句才会有效
24 db.commit()
25
26

```



5.影评爬取代码展示

数据不能直接使用

```

1
2 import csv
3 import requests
4 import xlwt
5 from base import Sqide
6 from lxml import etree
7 import pymysql
8 from pymongo import MongoClient
9 from elasticsearch import Elasticsearch
10
11 class Test(Sqide):
12     url = r'https://ssr1.scrape.center/'
13     url_list = []
14     data = []
15
16     def product(self):
17         response = requests.get(self.url)
18         content = response.content.decode('utf8')
19         html = etree.HTML(content)
20         urls = html.xpath('//div[@class="el-card item m-t is-hover-shadow"]//a[@class="name"]/@href')
21         for i in urls:
22             self.url_list.append('https://ssr1.scrape.center' + str(i))
23         print(self.url_list)
24
25     def consumer(self):
26         for i in self.url_list:
27             response = requests.get(i)
28             content = response.content.decode('utf8')
29             html = etree.HTML(content)
30             title = html.xpath('//h2[@class="m-b-sm"]/text()')
31             start = html.xpath(

```

```

25         '//div[@class="el-col el-col-24 el-col-xs-8 el-col-sm-4"]//p[@class="score m-t-md m-b
26         actors = html.xpath('//div[@class="el-card__body"]/p[@class="name text-center m-b-none m-
27         dir = {"title": title, "start": start, "actor": actors}
28         self.data.append(dir)
29
30     def save_csv(self):
31         header = self.data[0].keys()
32         # 写入文件test.csv,
33         with open('test.csv', 'w', encoding='utf8') as f:
34             # 提前预览列名, 当下面代码写入数据时, 会将其一一对应。
35             witer = csv.DictWriter(f, fieldnames=header)
36             # 写入列名
37             witer.writeheader()
38             # 写入字典中的数据
39             witer.writerows(self.data)
40
41     def save_excel(self):
42         workbook = xlwt.Workbook(encoding='utf8')
43         sheet1 = workbook.add_sheet('test')
44         header = list(self.data[0].keys())
45         for i, key in zip(range(len(header)), header):
46             sheet1.write(0, i, key)
47         for j in range(1, len(self.data) + 1):
48             for m, key in zip(range(len(header)), header):
49                 sheet1.write(j, m, self.data[j - 1][key])
50
51         workbook.save(r'C:\Users\Administrator\Desktop\test.xlsx')
52
53     def save_mongo(self):
54         mdb = MongoClient('127.0.0.1', 27017)
55         db = mdb.test['moive_info']
56         for i in self.data:
57             if isinstance(i, dict):
58                 db.insert_one(i)
59             else:
60                 print(i['title'], '写入失败')
61
62     def save_mysql(self):
63         db = pymysql.connect(host='127.0.0.1', user='root', password='123456', port=3306, database='t
64         cursor = db.cursor()
65         keys = ','.join(self.data[0].keys())
66         values = ','.join(['%s'] * len(self.data[0]))
67         sql = 'insert into moive_list(id,{key}) values(null,{values})'.format(key=keys, values=values)
68         for i in self.data:
69             value = list(i.values())
70             title = value[0][0]
71             start = value[1][0]
72             actor = value[2]
73             try:
74                 if cursor.execute(sql, (title, start, actor)):
75                     print(title, 'seccess')
76             except:
77                 db.rollback()
78             db.commit()
79         # cursor.execute(sql)
80
81     def practice_es(self):
82         es=Elasticsearch('http://127.0.0.1:9200/')
83         es.index(index='xxx',doc_type='he',id=1,body=self.data)

```

```

72 Test().product()
73 Test().consumer()
74 # Test().save_excel()
75 # Test().save_mysql()
76 # Test().save_mongo()
77 Test().practice_es()
78

```

79	
80	
81	
82	
83	
84	
85	
86	
87	
88	
89	
90	
91	
92	
93	
94	
95	
96	
97	
98	