

点击这里关注我们



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     writer = csv.DictWriter(f, fieldnames=header)
7     # 写入列名
8     writer.writeheader()
9     # 写入字典中的数据
10    writer.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
3 workbook = xlwt.Workbook(encoding='utf8')
4 # 新建一个excel书签, 名为test
5 sheet1 = workbook.add_sheet('test')
6 # 获取字段
7 header = list(self.data[0].keys())
8 # 使用for循环, 字段数的长度和字段
9 for i, key in zip(range(len(header)), header):
10 # excel写入字段名
11     sheet1.write(0, i, key)
12
13 #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
14 for j in range(1, len(self.data) + 1):
15     for m, key in zip(range(len(header)), header):
16         #写入数据
17         sheet1.write(j, m, self.data[j - 1][key])
18
19
20 #保存文件路径, 不用新建excel文件, 自动生成
21
22 workbook.save(r'C:\Users\Administrator\Desktop\test.xlsx')
23
24
25
26
27
28
29
30

```



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 #连接mongodb
2
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
21
22
23
24
25
26
27
28
29
30

```



4. 数据写入mysql

爬取的数据还是存放在类的self.data属性中

```

1 # 连接mysql数据库, 注: 注意指定database的名称
2 db = pymysql.connect(host='127.0.0.1', user='root', password='123456', port=3306, database='test')
3
4 #实例化一个游标对象
5 cursor = db.cursor()
6
7 #创建两个字段, keys为sql语句的字段的占位字段, values为数据的占位字符串
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

```

```

5 keys = ','.join(self.data[0].keys())
6 values = ','.join(['%s'] * len(self.data[0]))
#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
24
25
26

```



5.影评爬虫代码展示

数据不能直接使用

```

1 import csv
2 import requests
3 import xlwt
4 from base import Sqide
5 from lxml import etree
6 import pymysql
7 from pymongo import MongoClient
8 from elasticsearch import Elasticsearch
9
10 class Test(Sqide):
11     url = r'https://ssr1.scrape.center/'
12     url_list = []
13     data = []
14
15     def product(self):
16         response = requests.get(self.url)
17         content = response.content.decode('utf8')
18         html = etree.HTML(content)
19         urls = html.xpath('//div[@class="el-card item m-t is-hover-shadow"]//a[@class="name"]/@href')
20         for i in urls:
21             self.url_list.append('https://ssr1.scrape.center' + str(i))
22         print(self.url_list)
23
24     def consumer(self):
25         for i in self.url_list:
26             response = requests.get(i)
27             content = response.content.decode('utf8')
28             html = etree.HTML(content)
29             title = html.xpath('//h2[@class="m-b-sm"]/text()')
30             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 actors = html.xpath('//div[@class="el-card__body"]/p[@class="name text-center m-b-none m-dir = {"title": title, "start": start, "actor": actors} self.data.append(dir)
26
27     def save_csv(self):
28         header = self.data[0].keys()
29         # 写入文件test.csv,
30         with open('test.csv', 'w', encoding='utf8') as f:
31             # 提前预览列名, 当下面代码写入数据时, 会将其一一对应。
32             witer = csv.DictWriter(f, fieldnames=header)
33             # 写入列名
34             witer.writeheader()
35             # 写入字典中的数据
36             witer.writerows(self.data)
37
38     def save_excel(self):
39         workbook = xlwt.Workbook(encoding='utf8')
40         sheet1 = workbook.add_sheet('test')
41         header = list(self.data[0].keys())
42         for i, key in zip(range(len(header)), header):
43             sheet1.write(0, i, key)
44         for j in range(1, len(self.data) + 1):
45             for m, key in zip(range(len(header)), header):
46                 sheet1.write(j, m, self.data[j - 1][key])
47
48         workbook.save(r'C:\Users\Administrator\Desktop\test.xlsx')
49
50     def save_mongo(self):
51         mdb = MongoClient('127.0.0.1', 27017)
52         db = mdb.test['moive_info']
53         for i in self.data:
54             if isinstance(i, dict):
55                 db.insert_one(i)
56             else:
57                 print(i['title'], '写入失败')
58
59     def save_mysql(self):
60         db = pymysql.connect(host='127.0.0.1', user='root', password='123456', port=3306, database='t61         cursor = db.cursor()
62         keys = ','.join(self.data[0].keys())
63         values = ','.join(['%s'] * len(self.data[0]))
64         sql = 'insert into moive_list(id,{key}) values(null,{values})'.format(key=keys, values=values)
65         for i in self.data:
66             value = list(i.values())
67             title = value[0][0]
68             start = value[1][0]
69             actor = value[2]
70             try:
71                 if cursor.execute(sql, (title, start, actor)):
72                     print(title, 'seccess')
73             except:
74                 db.rollback()
75             db.commit()
76             # cursor.execute(sql)
77
78     def practice_es(self):
79         es=Elasticsearch('http://127.0.0.1:9200/')
80         es.index(index='xxx',doc_type='he',id=1,body=self.data)
81
82     Test().product()
83     Test().consumer()
84     # Test().save_excel()
85     # Test().save_mysql()
86     # Test().save_mongo()
87     Test().practice_es()
```

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

