

由于自己能力有限，因此目前只实现了 12306 验证码校验和查票功能

一、查票功能(关键代码如下):

```
def query(self, from_station, to_station, date):
    '''余票查询'''
    fromstation = self.station_name(from_station)
    tostation = self.station_name(to_station)
    url =
'https://kyfw.12306.cn/otn/leftTicket/query?leftTicketDTO.train_date={}&leftTicketDTO.from_s
tation={}&leftTicketDTO.to_station={}&purpose_codes=ADULT'.format(
    date, fromstation, tostation)
    # 学生票查询: url =
'https://kyfw.12306.cn/otn/leftTicket/query?leftTicketDTO.train_date={}&leftTicketDTO.from_s
tation={}&leftTicketDTO.to_station={}&purpose_codes=0X00'.format(date, fromstation, tostation)
    try:
        html = requests.get(url, headers=self.headers, verify=False).json()
        result = html['data']['result']
        if result == []:
            print('很抱歉,没有查到符合当前条件的列车!')
            exit()
        else:
            print(date + from_station + '-' + to_station + ' 查询成功!')
            # 打印出所有车次信息
            num = 1 # 用于给车次编号,方便选择要购买的车次
            for i in result:
                info = i.split('|')
                if info[0] != '' and info[0] != 'null':
                    print(str(num) + '.' + info[3] + ' 车次还有余票:')
                    print(' 出发时间:' + info[8] + ' 到达时间:' + info[9] + ' 历时多久:' + info[10]
+ ' ', end='')
                    seat = {21: '高级软卧', 23: '软卧', 26: '无座', 28: '硬卧', 29: '硬座', 30:
'二等座', 31: '一等座', 32: '商务座',
                        33: '动卧'}
                    from_station_no = info[16]
                    to_station_no = info[17]
                    for j in seat.keys():
                        if info[j] != '无' and info[j] != '':
                            if info[j] == '有':
                                print(seat[j] + ':有票 ', end='')
                            else:
                                print(seat[j] + ':有' + info[j] + ' 张票 ', end='')
                    print('\n')
                    elif info[1] == '预订':
                        print(str(num) + '.' + info[3] + ' 车次暂时没有余票')
```

```
        elif info[1] == '列车停运':
            print(str(num) + '.' + info[3] + '车次列车停运')
        elif info[1] == '23:00-06:00 系统维护时间':
            print(str(num) + '.' + info[3] + '23:00-06:00 系统维护时间')
        else:
            print(str(num) + '.' + info[3] + '车次列车运行图调整, 暂停发售')
            num += 1

    return result
except:
    print(' 查询信息有误!请重新输入!')
    exit()
```

```
def select():
    '''查票函数'''
    # 用户输入购票信息:
    from_station = input(' 请输入您要购票的出发地(例:北京):')
    # from_station = '北京'
    to_station = input(' 请输入您要购票的目的地(例:上海):')
    # to_station = '上海'
    date = input(' 请输入您要购票的乘车日期(例:2019-03-06):')
    # date = '2019-05-15'
    # 余票查询
    query = Leftquery()
    result = query.query(from_station, to_station, date)
```

运行效果部分截图:

```
12306 x
C:\Users\Lenovo\AppData\Local\Programs\Python\Python35\python.exe D:/爬虫/Spiders/12306.py
*****12306购票*****
请输入您要购票的出发地(例:北京):温州
请输入您要购票的目的地(例:上海):杭州
请输入您要购票的乘车日期(例:2019-03-06):2019-07-01
2019-07-01温州-杭州查询成功!
1. G590车次还有余票:
出发时间:06:14 到达时间:09:11 历时多久:02:57 商务座:有2张票 二等座:有票 一等座:有13张票

2. G7504车次还有余票:
出发时间:06:29 到达时间:09:17 历时多久:02:48 商务座:有票 二等座:有票 一等座:有票

3. G7482车次还有余票:
出发时间:06:37 到达时间:10:03 历时多久:03:26 商务座:有8张票 二等座:有票 一等座:有20张票

4. G7330车次还有余票:
出发时间:06:38 到达时间:09:04 历时多久:02:26 商务座:有13张票 二等座:有票 一等座:有票

5. G7572车次还有余票:
出发时间:06:43 到达时间:09:55 历时多久:03:12 二等座:有票 一等座:有票

6. K2906车次还有余票:
出发时间:07:00 到达时间:13:28 历时多久:06:28 软卧:有18张票 硬卧:有票 硬座:有17张票

7. G7674车次还有余票:
```

二、验证码校验部分(假设用户名和密码均正确)

```
def showing(self):
    '''显示验证码图片'''
    global req
    html_pic = req.get(self.url_pic, headers=self.headers).content
    open('pic.jpg', 'wb').write(html_pic)
    img = mpimg.imread('pic.jpg')
    plt.imshow(img)
    plt.axis('off')
    plt.show()

def captcha(self, answer_num):
    '''填写验证码'''
    answer_sp = answer_num.split(',')
    answer_list = []
    an = {'1': (31, 35), '2': (116, 46), '3': (191, 24), '4': (243, 50), '5': (22, 114), '6': (117, 94),
```

```

        '7': (167, 120), '8': (251, 105)}
for i in answer_sp:
    for j in an.keys():
        if i == j:
            answer_list.append(an[j][0])
            answer_list.append(',')
            answer_list.append(an[j][1])
            answer_list.append(',')

s = ''
for i in answer_list:
    s += str(i)
answer = s[:-1]
# 验证验证码
form_check = {
    'answer': answer,
    'login_site': 'E',
    'rand': 'sjrand',
    '_': str(int(time.time() * 1000))
}

global req
html_check = req.get(self.url_check, params=form_check, headers=self.headers).json()
print(html_check)
if html_check['result_code'] == '4':
    print('验证码校验成功!')
else:
    print('验证码校验失败!')
    exit()

def ver_code():
    login = Login()
    #login.get_rail_deviceid()
    login.showing()
    # 填写验证码
    print(' ===== ')
    print(' 根据打开的图片识别验证码后手动输入，输入正确验证码对应的位置 ')
    print(' -----')
    print('      1 | 2 | 3 | 4 ')
    print(' -----')
    print('      5 | 6 | 7 | 8 ')
    print(' ----- ')
    print(' ===== ')
    answer_num = input(' 请填入验证码(序号为1~8,中间以逗号隔开,例:1,2):')
    login.captcha(answer_num)
    #login.login(answer_num)

```

运行效果:

当输入用户名和密码(假设用户名和密码均正确)后,在进行验证码校验时,会自动弹出一张图片识别验证码(如下图所示),根据打开的图片识别验证码后手动输入正确验证码对应的位置,即可校验成功。



```
12306 x
C:\Users\Lenovo\AppData\Local\Programs\Python\Python35\python.exe D:/爬虫/Spiders/12306.py
*****12306购票*****
请输入用户名: 15558975317
请输入密码: 12345678

=====
      根据打开的图片识别验证码后手动输入, 输入正确验证码对应的位置
      -----
          1 | 2 | 3 | 4
      -----
          5 | 6 | 7 | 8
      -----

=====
请输入验证码(序号为1~8, 中间以逗号隔开, 例:1, 2): 1, 4
{'result_code': '4', 'result_message': '验证码校验成功'}
验证码校验成功!
```

注: 本次作业由于是以 12306 为主题, 所以没在期中作业的基础上继续往下做(django 部分可看期中作业), 且自己水平不好, 望老师体谅。