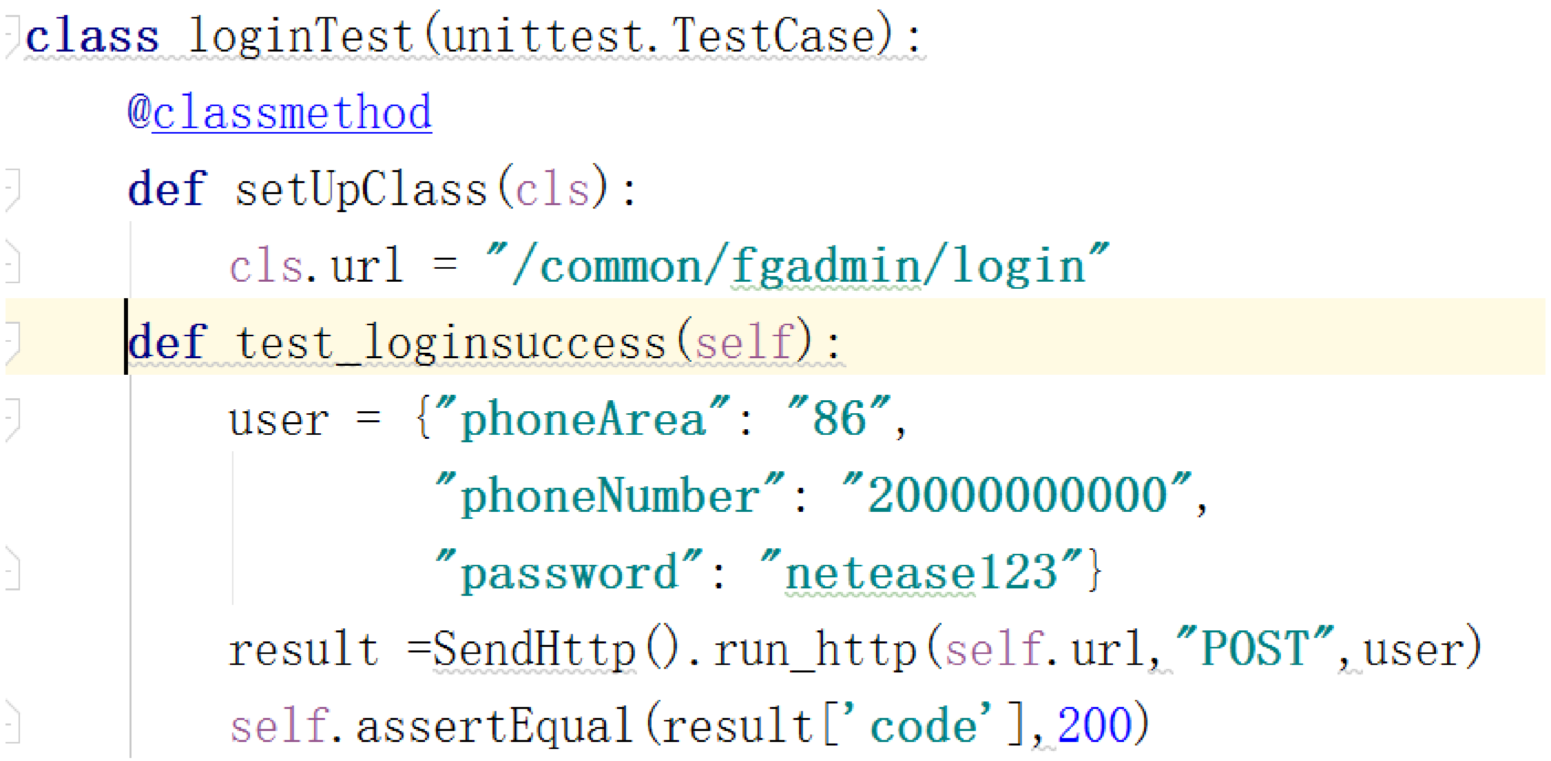
基于Python的接口框架设计

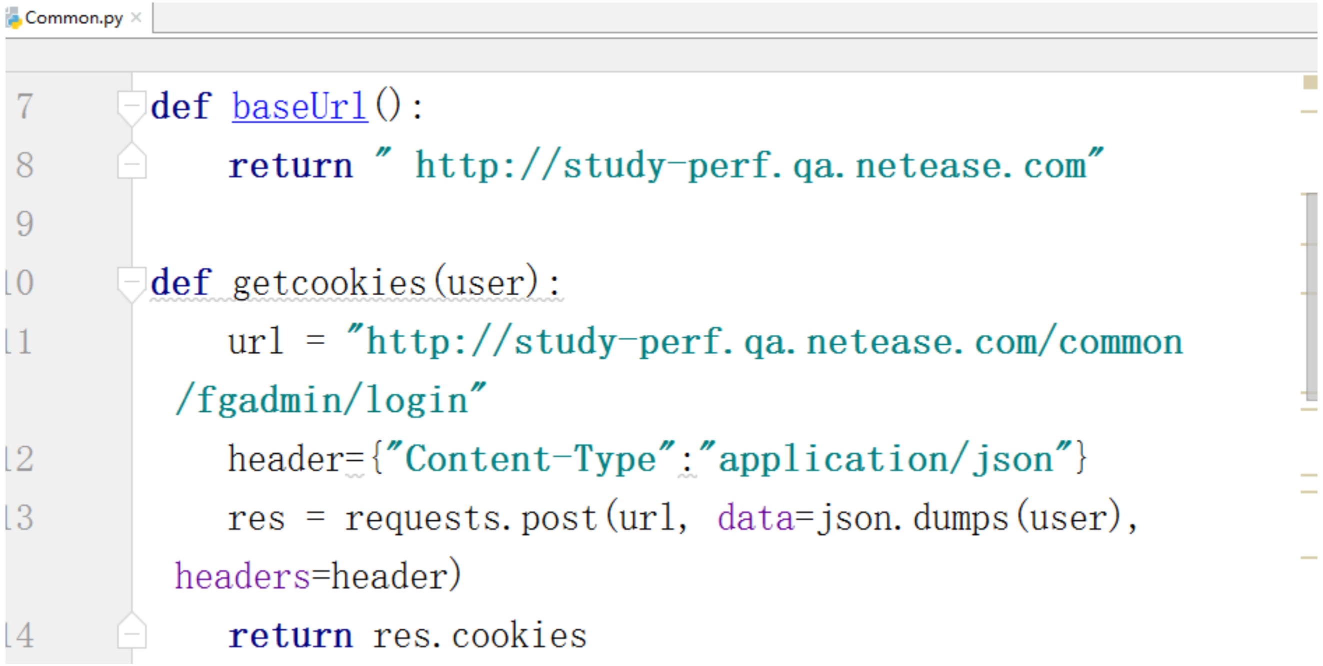
unittest和request重构封装



```python
class loginTest(unittest.TestCase):
    @classmethod
    def setUpClass(cls):
        cls.url = "/common/fgadmin/login"
    def test_loginsuccess(self):
        user = {"phoneArea": "86",
                "phoneNumber": "20000000000",
                "password": "netease123"}
        result =SendHttp().run_http(self.url,"POST",user)
        self.assertEqual(result['code'],200)
```

封装获取常量方法

1、存储域名
2、存储获得cookie方法

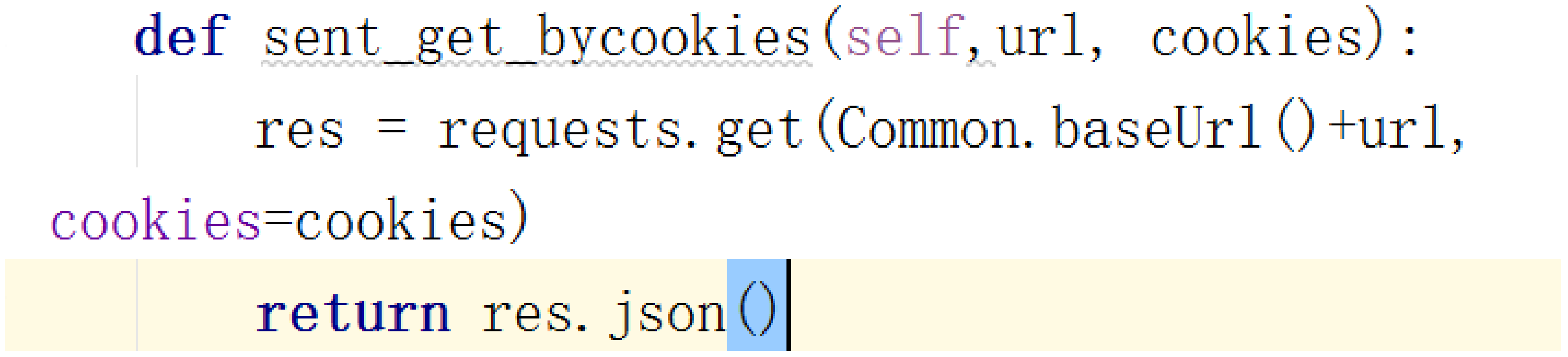


Common.py

```python
def baseUrl():
    return " http://study-perf.qa.netease.com"

def getcookies(user):
    url = "http://study-perf.qa.netease.com/common/fgadmin/login"
    header={"Content-Type":"application/json"}
    res = requests.post(url, data=json.dumps(user), headers=header)
    return res.cookies
```

cookie的获取



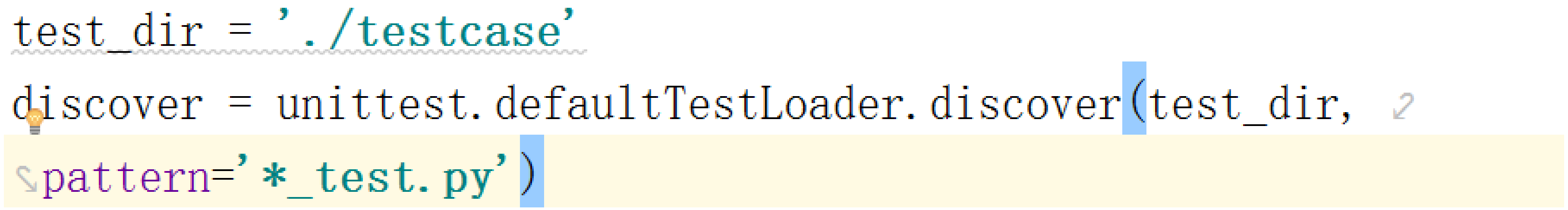
```
def sent_get_bycookies(self,url, cookies):
        res = requests.get(Common.baseUrl()+url,
cookies=cookies)
        return res.json()
```

unittest中case的管理及运用

- 每个接口对应响应的.py文件，如 login_test.py
- 每个文件中需要包含通过和失败的测试用例
- 包含断言

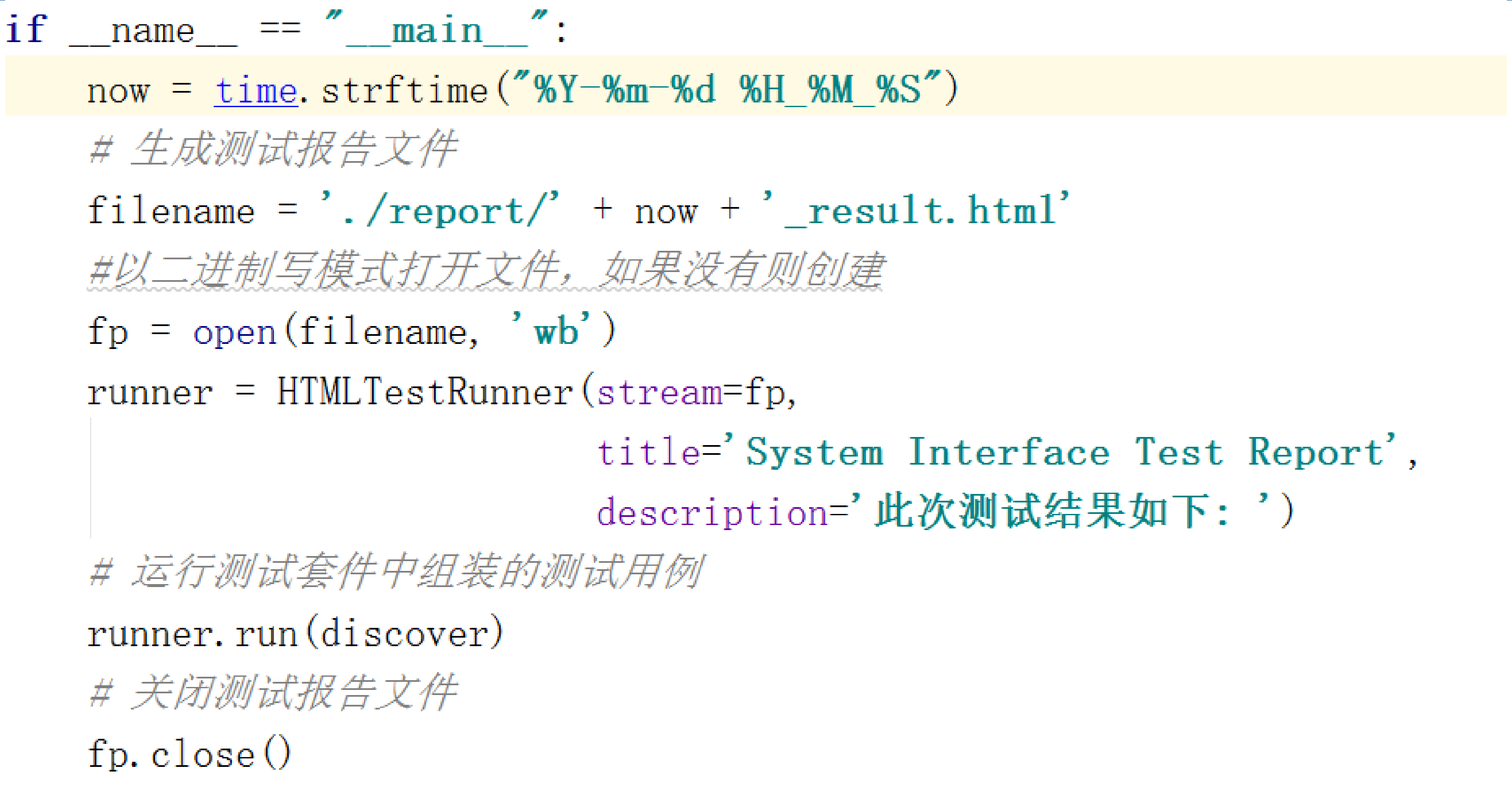
unittest和HTMLTestRunner结合生成报告

- 获取指定路径的.py文件



```
test_dir = './testcase'
discover = unittest.defaultTestLoader.discover(test_dir,
pattern='*_test.py')
```

unittest和HTMLTestRunner结合生成报告



```python
if __name__ == "__main__":
    now = time.strftime("%Y-%m-%d %H_%M_%S")
    # 生成测试报告文件
    filename = './report/' + now + '_result.html'
    #以二进制写模式打开文件，如果没有则创建
    fp = open(filename, 'wb')
    runner = HTMLTestRunner(stream=fp,
                            title='System Interface Test Report',
                            description='此次测试结果如下：')
    # 运行测试套件中组装的测试用例
    runner.run(discover)
    # 关闭测试报告文件
    fp.close()
```

python操作excel获得内容

- **import** xlrd
- **def** readExcel(filePath,index):

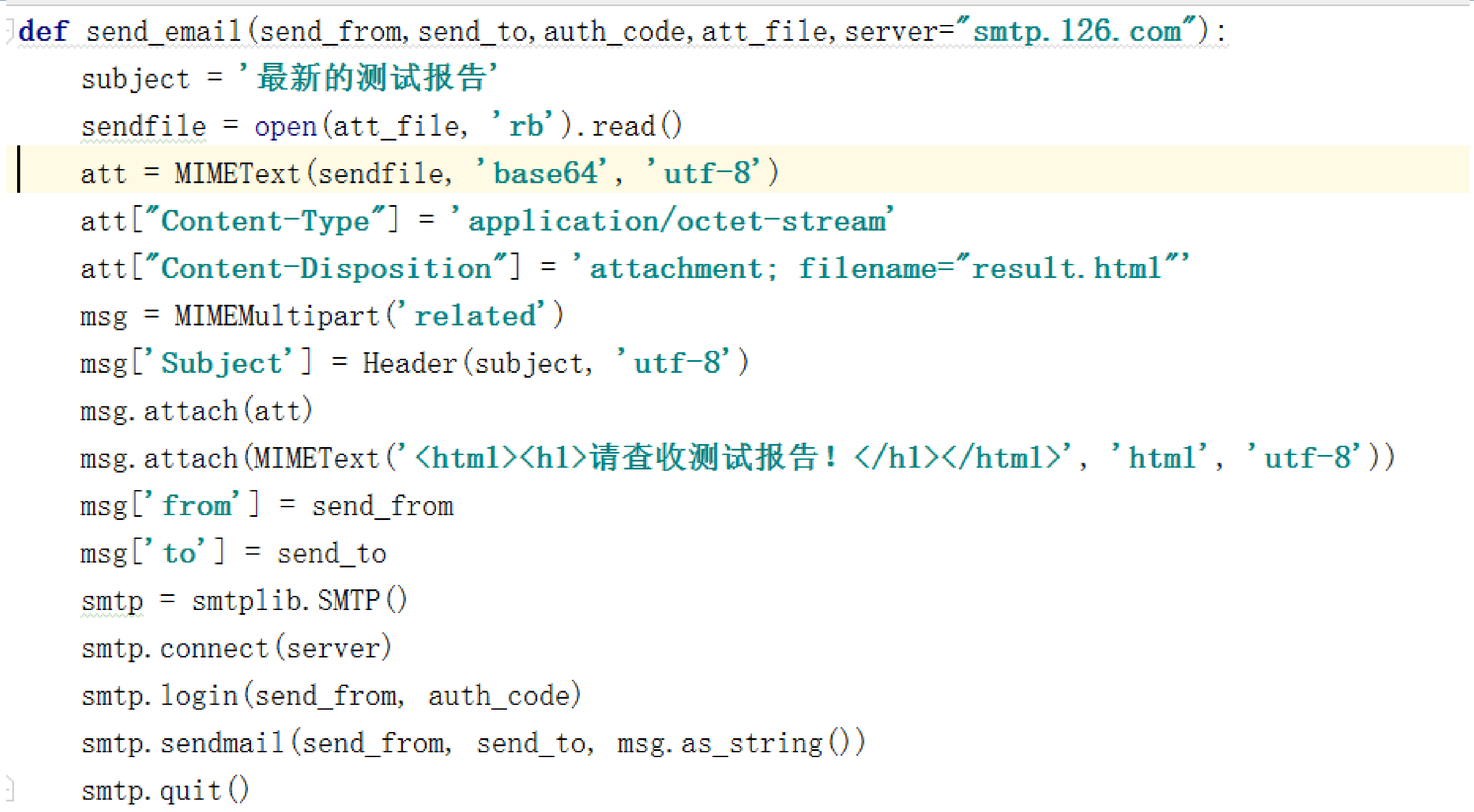
```
    workbook = xlrd.open_workbook(filePath)
    table = workbook.sheet_by_index(index)
    return  table
```

python操作excel获得内容

```
def test_login_by_data(self):
    sheet = dp.readExcel(r"D:\demo\userdata.xlsx", 0)

    for i in range(sheet.nrows):
        user = {"phoneArea": "86",
              "phoneNumber": sheet.cell_value(i, 0),
              "password": "netease123"}
        result = SendHttp().run_http(self.url, "POST",user)
        print(result)
```

构建发送邮件服务



```python
def send_email(send_from,send_to,auth_code,att_file,server="smtp.126.com"):
    subject = '最新的测试报告'
    sendfile = open(att_file, 'rb').read()
    att = MIMEText(sendfile, 'base64', 'utf-8')
    att["Content-Type"] = 'application/octet-stream'
    att["Content-Disposition"] = 'attachment; filename="result.html"'
    msg = MIMEMultipart('related')
    msg['Subject'] = Header(subject, 'utf-8')
    msg.attach(att)
    msg.attach(MIMEText('<html><h1>请查收测试报告！</h1></html>', 'html', 'utf-8'))
    msg['from'] = send_from
    msg['to'] = send_to
    smtp = smtplib.SMTP()
    smtp.connect(server)
    smtp.login(send_from, auth_code)
    smtp.sendmail(send_from, send_to, msg.as_string())
    smtp.quit()
```

python

requests 接口测试， 发送http请求

unittest 管理测试用例

HTMLTestRunner 生成测试报告

xlrd 获取excel的文件,实现数据驱动模式

smtplib 发送邮件

httpclient 接口测试，发送http请求

testng 管理测试用例,生成测试报告

freemarker/reportNG美化测试报告

apache poi 获取excel的文件,实现数据驱动模式

javamail 发送邮件