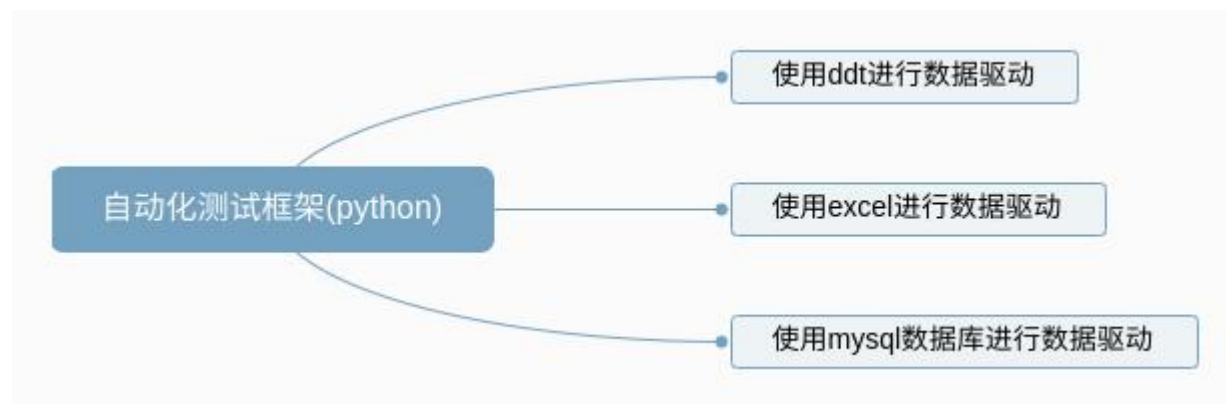


自动化测试框架 (python)-数据驱动测试

讲师：芳姐

| 时间：2018-6-23



使用Excel进行数据驱动测试

一、测试逻辑

1. 打开erp登录页面，从Excel文件中读取用户名和密码作为输入关键词
2. 在用户名和密码输入框输入读取出的关键词
3. 点击登录按钮
4. 断言登录后的结果页面中是否出现Excel文件中提供的预期内容，包含则认为测试执行成功，否则失败

二、环境准备

1. 安装openpyxl模块，Windows在cmd里面输入,linux 在命令行终端输入
2. 输入pip3 install openpyxl==2.3.3，进行安装Python解析Excel2007及以上版本的模块
3. 安装成功如下：

```
Downloading https://files.pythonhosted.org/packages/22/28/a99c42aea746e18382a0
9fb36f64c1c1f04216f41797f2f0fa567da11388/et_xmlfile-1.0.1.tar.gz
Installing collected packages: jdcal, et-xmlfile, openpyxl
  Running setup.py install for et-xmlfile ... done
  Running setup.py install for openpyxl ... done
Successfully installed et-xmlfile-1.0.1 jdcal-1.4 openpyxl-2.3.3
```

三、测试数据准备

在本地磁盘E:\python\ERP_selenium UI自动化(Python)\第九课目录中新建一个“测试数据.xlsx”，工作表名为“用户登录”的Excel文件，“用户登录”工作表内容如下：

A	B	C	D
序号	用户名	密码	期望结果
1	defang1	123	德芳理财

四、实例代码

```
from openpyxl import load_workbook
class ParseExcel(object):
    def __init__(self, excelPath, sheetName):
        # 将要读取的Excel加载到内存
        self.wb = load_workbook(excelPath)
        # 通过工作表名称获取一个工作表对象
        self.sheet = self.wb.get_sheet_by_name(sheetName)
        # 获取工作表中存在数据的区域的最大行号
        self.maxRowNum = self.sheet.max_row
```

```
def getDatasFromSheet(self):  
    # 用于存放工作表中读取出来的数据  
    dataList = []  
    # 因为工作表中的第一行是标题行，所以需要去掉  
    for line in self.sheet.rows[1:]:  
        # 遍历工作表中数据区域的每一行  
        # 将每行中各个单元格的数据取出存于列表tmpList中  
        # 然后再将存放一行数据的列表添加到最终数据列表dataList中  
        tmpList = []  
        tmpList.append(line[1].value)  
        tmpList.append(line[2].value)  
        tmpList.append(line[3].value)  
        dataList.append(tmpList)  
    return dataList
```

```
if __name__ == '__main__':  
    excelPath = 'E:\\python\\ERP_selenium UI自动化(Python)\\第九课\\测试数据.xlsx'  
    sheetName = "用户登录"  
    pe = ParseExcel(excelPath, sheetName)  
    for i in pe.getDatasFromSheet():  
        print(i[0], i[1])
```

代码见：

ddt数据驱动测试/DataTest.py

日志：

```
logging.basicConfig(  
    # 日志级别  
    level=logging.INFO,  
    # 日志格式  
    # 时间，代码所在文件名，代码行号，日志级别名字，日志信息  
    format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelname)s %(message)s',  
    # 打印日志的时间  
    datefmt='%a, %Y-%m-%d %H:%M:%S',  
    # 日志文件存放的目录（目录必须存在）及日志文件名  
    filename='E:\\Python\\DataDrivenProject\\report.log',  
    # 打开日志文件的方式  
    filemode='w'  
)
```

```
excelPath = 'E:\\python\\ERP_selenium UI自动化(Python)\\第九课\\测试数据.xlsx'  
sheetName = "用户登录"  
# 创建ParseExcel类的实例对象  
excel = ParseExcel(excelPath, sheetName)
```

```
@ddt.ddt  
class TestDemo(unittest.TestCase):  
  
    def setUp(self):  
  
        self.driver = webdriver.Chrome(executable_path="E:\\Python36\\chromedriver")
```



```
@ddt.data(* excel.getDatasFromSheet())
def test_dataDrivenByFile(self, data):
    username, password, expectData = tuple(data)
    url = "http://10.1.2.211:8080/login.jsp"
    # 访问登录首页
    self.driver.get(url)
    # 将浏览器窗口最大化
    self.driver.maximize_window()
    print(username, password, expectData)

    # 设置隐式等待时间为10秒
    self.driver.implicitly_wait(10)

try:
    # 找到用户名和密码输入框，并输入测试数据
    self.driver.find_element_by_xpath("//*[@name='username']").send_keys(username)
    self.driver.find_element_by_xpath("//*[@name='password']").send_keys(password)
    # 找到登录按钮，并单击
    self.driver.find_element_by_xpath("//*[@class='login-btn']").click()
    time.sleep(3)
```

```
# 断言期望结果是否出现在页面源代码中
self.assertTrue(expectData in self.driver.page_source)
except NoSuchElementException as e:
    print(e)
    logging.error("查找的页面元素不存在，异常堆栈信息： " + str(traceback.format_exc()))

except AssertionError as e:
    print(e)
    logging.info("登录'%s',期望'%s',失败" % (username, expectData))

except Exception as e:
    print(e)
    logging.error("未知错误，错误信息： " + str(traceback.format_exc()))

else:
    logging.info("登录'%s',期望'%s',成功" % (username, expectData))
```

使用MySQL进行数据驱动测试

- 1、日志信息和之前一样的
- 2、MySQL的信息

```
class MyMySQL(object):  
    def __init__(self, host, port, dbName, username, password, charset):  
        # 进行数据库初始化  
        # dbInit = DataBaselInit(host, port, dbName, username, password, charset)  
        self.conn = pymysql.connect(  
            host=host,  
            port=port,  
            db=dbName,  
            user=username,  
            passwd=password,  
            charset=charset  
        )  
        self.cur = self.conn.cursor()
```

```
def getDataFromDataBases(self):  
    # 从testdb表中获取需要的测试数据  
    # Movie_name作为用户名， Movie_property作为密码， excpect_result作为预期结果  
    self.cur.execute("select Movie_Name,Movie_Property,Excpect_result from testdata; ")  
    # 从查询预期取回所有查询结果  
    datasTuple = self.cur.fetchall()  
    return datasTuple  
  
def closeDatabase(self):  
    # 数据库后期清理工作  
    self.cur.close()  
    self.conn.commit()  
    self.conn.close()
```

3、连接MySQL

```
def getTestDatas():  
    db = MySQL(          
        host="10.1.2.71",  
        port=3306,  
        dbName="testdb",  
        username="root",  
        password="testjfz",  
        charset="utf8"  
    )  
  
    #从数据库测试表中获取测试数据  
    testData = db.getDataFromDataBases()  
    #关闭数据库连接  
    db.closeDatabase()  
    return testData
```

```
@ddt.data(*getTestDatas())  
def test_dataDrivenByFile(self, data):  
    username, password, expectData = tuple(data)
```

日志信息:

```
Tue, 2018-06-12 20:41:03 dataDrivenByMQL.py[line:86] INFO 登录'defang1',期望'德芳理财',失败  
Tue, 2018-06-12 20:41:17 dataDrivenByMQL.py[line:86] INFO 登录'defang2',期望'德芳客服',失败  
Tue, 2018-06-12 20:41:36 dataDrivenByMQL.py[line:93] INFO 登录'defang3',期望'德芳运维',成功
```

作业：

1、使用Excel进行数据驱动

2、使用MySQL进行数据驱动



Thanks!

科 技 提 升 投 资 品 质