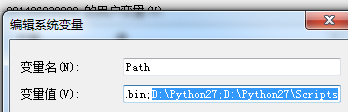
# 安装

* 安装python版本python-2.7.9，且该版本含有easy\_install 和pip，便于安装外部包，3以上的版本，不太稳定，
* 安装打开python语言的工具，pycharm-community-4.0.6.exe
* 配置环境变量，在path中添加python的安装路径，和python的script(便于进入cmd下可以直接easy\_install LIB包)如：



# Suds的基本使用与详解

Easy\_install suds外部包，suds.client对webservice不同的wsdl的规范，有相应的解析。

* Client

**from** suds.client **import** Client #引入suds url **=** “htp://pre-pdf.creditease.corp/services/ceBorrowingProducts?wsdl”

client **=** Client(url)

print client #打印生成的client的信息





Ns0:命名空间

Mothods为wsdl下的所有接口

Type:为Mothods方法下所有的对象

* Factory与service
* 当接口中的参数为普通类型，如getMobile(xs:string arg0, )

result=client.service.getMobile("{'Signage':'NEIWANG\_eehErjllPxSI90fYRW79Fvp4','empNo':'201210290131'}"),result为接口返回的结果。

* 当接口中的参数为对象，即为复杂的参数，就需要用到factory来创建该对象，并给该对象附值。
* Methods (1):

ceContractInfoForProduct(contractInfoForProductReq，contractInfoForProductReq)

**contractInfoForProductReq**为对象，所以需要**contractInfoForProductReq**=client.factory.create(**contractInfoForProductReq**)创建该对象，print **contractInfoForProductReq 如下：**

(contractInfoForProductReq){

lenderType = None

productInfoId = None

}

为该参数附上值:

productDiscountReq.departmentID= data[0][**"departmentID"**]  productDiscountReq.productInfoId= data[0][**"productInfoId"**] productDiscountReq.productCategory= data[0][**"productCategory"**] productDiscountReq.queryDate= data[0][**"queryDate"**] productDiscountReq.auditDate= data[0][**"auditDate"**] productDiscountReq.periodCount=data[0][**"periodCount"**] result=client.service.queryProductDiscount(productDiscountReq)

result为接口返回的结果。

* Factory创建的对象，生成的对象还包含对象，如：

queryInfo(queryInfoReq arg0, )

queryInfoReq= client.factory.create(**'queryInfoReq'**)

print queryInfoReq，如下结果

(queryInfoReq){

body =

(queryInfoReqBody){

remark1 = None

remark2 = None

remark3 = None

requestInfo = None

}

该对象还包含对象，再一次创建对象，如下：client.factory.create(**'queryInfoReq'**)，然后再对参数附上值

queryInfoReqBody.requestInfo= data[0][**"requestInfo"**]  
queryInfoReqBody.remark2= data[0][**"remark2"**]  
queryInfoReqBody.remark3= data[0][**"remark3"**]

result=client.service.queryInfo(**'queryInfoReq'**)

* 对象为命名空间

Methods (1):

openAccountCustomer(ns2:newCustomer arg0, xs:string arg1, xs:string arg2, )

ns2:newCustomer为命名空间，创建factory时，需要加上ns2,如下：

queryProductInfoIdReq=client.factory.create(**"ns2:newCustomer"**)

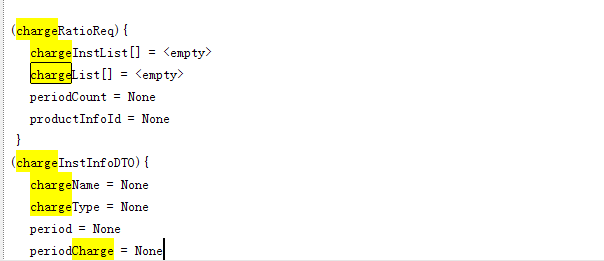
newCustomer .userDesc= “abc”

tui\_yingyebu.tui\_yingyebu=”ecd”

result=client.service.openAccountCustomer(queryProductInfoIdReq,”arg11”,”arg22”)

result为接口返回的结果。

* 创建对象问题，需要查看type下的值，如chargeInstList[]，该对象在type下是chargeInstInfoDTO,所以创建对象时候需要self.chargeInstList=self.client.factory.create('chargeInstInfoDTO')



# 接口系统各模块说明

* Config中url.py用来添加需要测试的wsdl的链接
* Data中用来防止每个系统参数化信息的excel表单
* Lib模块
* readexceldata.py，读取excel的方法，返回的是list组
* writeExcel.py，写入excel的方法
* test\_HTMLTestRunner.py，HTMLTestRunner.py生成报告的的方法
* utils.py，返回json格式，xml格式等公共方法
* TestCase模块
* Testbase,用于每个接口的数据初始化和生成client的初始化
* 以系统为单位，在每个系统下再进行各个接口的用例编写
* TestSuie ,用于组装各个系统的接口，用于批量运行，生成报告

# 编码规范

* 每个.py文件中，首行必须编码声明：*-\*- coding: utf-8 -\**
* 所有的测试用例都以”test\_\*”为命名
* 所有的测试用例都包含setup和teardown
* 所有的测试用例的参数必须后面加上注释
* 所有的测试用例必须写上断言。

# 代码问题总结

* 路径问题
* 编码问题
* Excel格式的参数化数据必须为文本格式，对于数据，用eval()的形式取值