# Thrift框架服务端接口自动化总结

**Windows下thrift安装**

1、下载thrift，下载地址：[http://archive.apache.org/dist/thrift/0.9.3/](http://archive.apache.org/dist/thrift/0.9.3/" \t "_blank)

2、在编写python的thrift代码时，需要先安装thrift module，下载路径：<https://pypi.python.org/pypi/thrift/0.9.1>

或者使用pip安装

3、安装thrift：在本地新建一个Thtift文件夹，将下载的thrift-0.x.x.exe重新命名为thrift.exe后放到Thtift文件夹下，将该路径添加到环境变量，检验是否安装成功：在命令行中输入：

thrift –version

即可以看到安装后的版本信息。

**Thrift使用**

第一步：定义接口

需要开发提供相应文件

e.g. ：user.thrift

内容如下：

e.g.：

1. service Parse {
2. string parseHtml2Xml(1:string html)
3. }

第二步：生成文件

在thrift接口所在目录执行thrift命令

thrift --gen py user.thrift

这步会在当前目录生成gen-py文件夹

PS：如果是用java，则用如下命令：

Linux环境：thrift --gen java user.thrift

windows环境: thrift-0.9.3.exe  --gen java user.thrift

**要注意的是，服务端和客户端使用的thrift版本要保持一致**

第三步：客户端开发

#coding = utf-8

from thrift import Thrift

#from thrift.transport import TSocket

from thrift.transport import TTransport

from thrift.protocol import TBinaryProtocol

from thrift.transport import THttpClient

#开发使用的thrift框架http请求，需要进入python在thrift包里找到合适的对应方法

sys.path.append('D:/thrift/gen-py') #根据自己环境路径写

import ttypes

import UserService

import time

import sys

import assertpy

class client(unittest.TestCase,conf\_sys):

#自定义类

def setUp(self):

transport = self.transport

transport.open()

def teardown(self):

transport = self.transport

transport.close()

def test\_login(self):

#main

client = self.client

a = self.a

try:

assert client.login(a).login\_token is not None

except BaseException , ex :

print ex

e = str(ex)

try:

assertpy.assert\_that(e).contains('HealthServiceException')

except:

try:

assertpy.assert\_that(e).contains('HTTP')

except:

raise TypeError('ICE Error')

else:

print ex.message

raise TypeError('HTTP ERROR')

else:

print(e)

1.用命令生成的文件，部分会由于语法或者路径原因导致语法错误或者找不到文件，

需要自己手动修改

2.参数资源池需要根据实际情况进行配置