火龙果·整理
uml.org.cn

接口测试及接口测试自动化

目录

火龙果·整理
uml.org.cn

Chapter 1 接口测试介绍

Chapter 2 HTTPie接口测试工具

Chapter 3 Postman接口测试工具

Chapter 4 RF自动化接口测试

Chapter 5 Python自动化接口测试

火龙果·整理
uml.org.cn

Chapter 1 接口测试介绍

1.1 什么是接口测试

1.2 HTTP常见概念

1.3 HTTP接口测试

1.4 数据格式

1.5 一个接口设计的例子

1.6 Fiddler

1.1 什么是接口测试

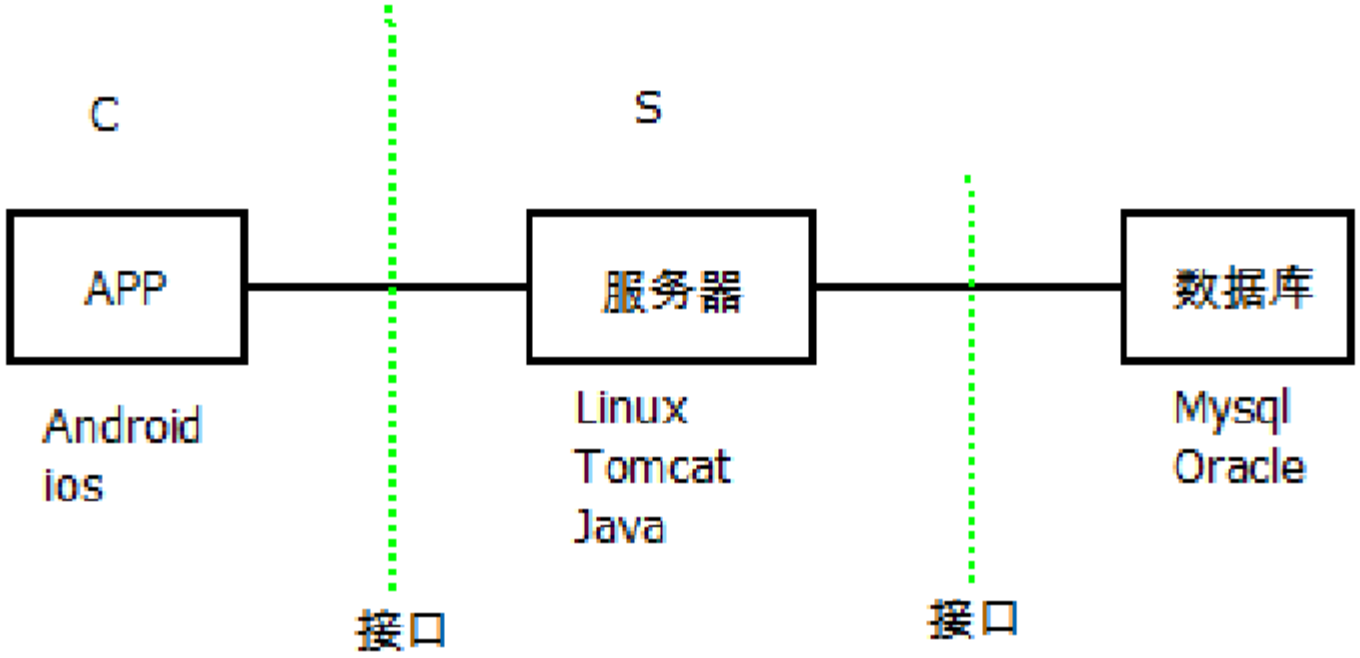
火龙果·整理
uml.org.cn

简介： 软件测试中所说的接口又称为API，接口测试是指用于与外部系统交互接口及软件系统内部模块/组件或子系统之间的内部接口测试。例如：

1、系统与系统之间的调用，比如银行会提供接口供支付宝网站调用，支付宝会提供接口给淘宝调用；

2、上层服务对下层服务的调用，比如服务层会调用数据库访问层的接口，而应用层又会调用服务层提供的接口；

3、服务之间的调用，比如注册用户时，会先调用用户查询的服务，查看该用户是否已经注册。



1.1 什么是接口测试

接口测试技术：

1. 工具和技术选型：接口上传输数据的内容格式协议及数据的传输承载协议类型决定该接口测试需要使用的工具和技术，协议通常包括自定义协议类型及标准协议类型，常见的标准/半标准协议类型包括:
   JSON/XML on REST/HTTP
   JSON/XML on SOAP/HTTP
   JMS on MQ
   ProtoBuf
   Thrift等

接口都由一组协议组成，定义会话层，表示层以及应用层的格式、内容等！

2. 功能测试方法：根据接口上下层之间的关系，需要选择不同的测试方法
   如果要测试下游的服务器接口，可以采用调用的方法进行测试；
   如果要对APP接口进行测试，需要使用打桩(stub)或者moco的方法进行测试；

3. 接口测试问题定位方法：接口测试通常的检查点包括：

检查点	检查方法
输入/输出数据的格式和内容	抓包，数据解析
程序内部状态机迁移	系统日志，调试工具
数据持久化	数据库查询
性能	RT，QPS，TPS，RPS
安全	代码注入，Fuzzing

1.1 什么是接口测试

软件HTTP接口举例： HTTP协议属于会话层协议，具有成熟完善，易开发易调试易理解，行业支持力度大等特点，是目前比较常见的接口协议，比如通过HTTP协议传输的经度和纬度数据查询所在城市的接口:

```
F:\>http http://apis.juhe.cn/geo/ lng==119.9772857 lat==27.327578 type==1 key==f
d49cfa4c68ebdd8dcb94ce6202dd230
```


```
{
    "error_code": 0,
    "reason": "Successed!",
    "result": {
        "address": "福建省宁德市福鼎市",
        "business": "",
        "citycode": 192,
        "ext": {
            "adcode": "350982",
            "city": "宁德市",
            "country": "中国",
            "country_code": 0,
            "direction": "",
            "distance": "",
            "district": "福鼎市",
            "province": "福建省",
            "street": "",
            "street_number": ""
        },
        "lat": "27.327578",
        "lng": "119.9772857",
        "type": "1"
    },
    "resultcode": "200"
}
```


http: 命令行接口测试工具
http://apis.juhe.cn/geo/ : URI,提供接口服务的网址
Lng:: 经度
Lat: 纬度

1.2 HTTP常见概念

HTTP接口常见概念： HTTP属于会话层协议，其下层协议是TCP协议，因此HTTP接口通常会有长连接，短连接的概念：

长连接：收发完数据后，不立刻关闭连接，可以多次收发数据包。
短连接：收发完数据后，立刻关闭连接。
心跳：长连接在没有数据通信时，定时发送数据包(心跳)，以维持连接状态。

统一资源定位符是对可以从互联网上得到的资源的位置和访问方法的一种简洁的表示，是互联网上标准资源的地址。互联网上的每个文件都有一个唯一的URL，它包含的信息指出文件的位置以及浏览器应该怎么处理它。

http://bornmoney.com/money/rmb?quantity=1000000
协议　主机　路径　查询参数

大概相当于：中文://深圳.广东.中国/福田区/青年大厦/5楼/公司?名字=门道

URL编/解码：URL中的字符只能是英文及少数几种标点符号，其他比如中文，特殊标点符号都需要经过转换，转换过程称为URL编/解码；

常见编码方式有
英文字符：ASCII码
非英文字符：UTF-8,GBK,Unicode等

```
+ 转义符为 %2B
空格 转义符为 + 或 %20
/ 转义符为 %2F
? 转义符为 %3F
% 转义符为 %25
# 转义符为 %23
& 转义符为 %26
= 转义符为 %3D
门道 转义为 %E9%97%A8%E9%81%93
```

1.2 HTTP常见概念

HTTP常用请求方法:

方法	描述
POST	**增加(合同，订单，用户，任务等)**
DELETE	**删除**
PUT	**修改**
GET	**查询**
OPTIONS	允许客户端查看服务器的性能。
TRACE	回显服务器收到的请求，主要用于测试或诊断。
HEAD	类似于get请求，只不过返回的响应中没有具体的内容，用于获取报头

HTTP常见响应状态码：

状态码	意义
200	OK
201	创建成功
204	请求成功无返回内容
302	临时移动
304	没有修改
400	错误请求
401	没有授权
403	禁止访问
404	资源不存在
405	方法不支持

1.3 HTTP接口测试

火龙果·整理
uml.org.cn

HTTP接口测试常用测试点:

1，功能测试：
- CRUD，增删改查功能测试
- 业务逻辑测试
- 删除/恢复测试

2，异常注入测试：
- 测试参数的无效等价类
- 重复增删改查

3，安全测试：
- HTTPS
- 各种认证/加密算法的测试
- 代码注入/SQL注入/JSON注入

4，性能测试，
- 并发用户数，RT，QPS，TPS，CPU，IO等
- 满配测试
- 批量操作

HTTP接口常用测试方法及工具:

1，手工功能测试：
- postman/jmeter/soapui等
- httpie/curl/httpsender等

2，自动化测试：
- Java+TestNG
- Python+pytest
- RobotFramework

3，安全测试：
- openssl
- httpie/postman
- SQLMAP/BurpSuite/ZAP等

4，性能测试，包括
- Jmeter,Loadrunner等
- ab,tsung,siege等

5，抓包工具
- Fiddler
- Wireshark/Tcpdump

1.3 HTTP接口测试

接口测试流程:

需求评审->接口定义->用例设计->测试执行->结果分析

注：接口一般是开发人员定义的

例如： 使用QQ号自动登陆的接口定义如右：

【QQ登录】API文档

分组名称	适用平台	API接口	功能说明
访问我的高级资料	仅网站	get_user_info	在网站上显示登录用户的昵称、头像、性别
访问我的基础资料	网站，移动	get_simple_userinfo	在网站上显示登录用户的昵称、头像
发表分享到QQ空间	网站，移动	add_share	发表一个网页分享，分享应用中的内容给好友

3.1请求说明

url	https://graph.qq.com/user/get_user_info
支持验证方式	oauth2.0
格式	JSON
http请求方式	GET

1.4 数据格式

会话层之上是表示层和应用层，定义用户数据的格式及用户数据的内容。用户数据的格式会有很多，比如视频，音频，文本、压缩文件等。而在HTTP接口测试中，用户数据通常会以文本的格式传输，采用JSON或者XML协议：

JSON(JavaScript Object Notation, JS 对象标记)相比于XML(可扩展标记语言)，其语法比较简洁，传输及处理时效率较高。

JSON格式

```
{
    "bookstore":{
            "book":{
                    "title":"Harry Potter",
                    "lang":"eng",
                    "price":29.99
            },
            "book":{
                    "title":"Learning XML",
                    "lang":"eng",
                    "price":39.95
            }
    }
}
```

XML格式

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book>
    <title lang="eng">Harry Potter</title>
    <price>29.99</price>
  </book>
  <book>
    <title lang="eng">Learning XML</title>
    <price>39.95</price>
  </book>
</bookstore>
```

1.5 一个接口设计说明

HTTP Method	URI	Action
Get	http://[hostname]/todo/api/v1.0/tasks	获取所有 Tasks 的列表
Get	http://[hostname]/todo/api/v1.0/tasks/[task_id]	获取给定 Task Id 的任务内容
POST	http://[hostname]/todo/api/v1.0/tasks	创建一个新的 Task
PUT	http://[hostname]/todo/api/v1.0/tasks/[task_id]	根据 Task Id 更新一个已有的 Task
DELETE	http://[hostname]/todo/api/v1.0/tasks/[task_id]	根据 Task Id 删除一个已有的 Task

这个接口定义了通过HTTP接口管理一个task数据的方法，包含下面的内容：
Task的资源位置
查询Task内容的方法
增加/创建Task的方法
修改某个指定Task的方法
删除某个指定taskId的方法

1.6 Fiddler

火龙果·整理
uml.org.cn

Fiddler是目前最常用的HTTP抓包工具之一，它能够记录客户端和服务器之间的所有HTTP请求，可以针对特定的HTTP请求，分析请求数据、设置断点、调试web应用、修改请求的数据，甚至可以修改服务器返回的数据，功能强大，是常用调试工具

Fiddler Web Debugger

File Edit Rules Tools View Help GET /book

Replay Go Stream Decode Keep: All sessions Any Process Find Save Browse Clear Cache TextWizard

#	Result	Protocol	Host	URL	Body
1	200	HTTP	www.google.com.hk	/webhp?client=aff-1616d...	21,018
2	304	HTTP	www.google.com.hk	/images/nav_logo170_hr....	
3	304	HTTP	ssl.gstatic.com	/gb/images/b_8d5afc09.png	
4	304	HTTP	www.google.com.hk	/images/srpr/logo11w.png	
5	304	HTTP	www.google.com.hk	/xjs/_/js/k=xjs.s.en_US.E...	
6	304	HTTP	www.google.com.hk	/xjs/_/js/k=xjs.s.en_US.E...	
7	304	HTTP	www.google.com	/textinputassistant/tia.png	
8	204	HTTP	www.google.com.hk	/gen_204?v=3&s=webhp...	
9	304	HTTP	www.google.com.hk	/images/nav_logo170.png	
10	302	HTTP	cdv.firefoxchina.cn	/cdv/v1/hosts.json	160
11	200	HTTP	open.tuniu.com	/distributor/product/frefo...	1,94
12	200	HTTP	i.firefoxchina.cn	/newtab/mini.html	16,86
13	304	HTTP	i.firefoxchina.cn	/newtab/images/upload/2...	
14	304	HTTP	i.firefoxchina.cn	/newtab/images/upload/2...	
15	304	HTTP	i.firefoxchina.cn	/newtab/images/upload/2...	
16	304	HTTP	i.firefoxchina.cn	/newtab/images/upload/2...	
17	304	HTTP	i.firefoxchina.cn	/newtab/images/upload/2...	
18	304	HTTP	i.firefoxchina.cn	/newtab/images/upload/2...	
19	304	HTTP	i.firefoxchina.cn	/newtab/images/upload/2...	

Statistics Inspectors AutoResponder Composer Filters Log Timeline

```
Request Count:    1
Bytes Sent:       635        (headers:635; body:0)
Bytes Received:   208        (headers:208; body:0)

ACTUAL PERFORMANCE
--------------
ClientConnected:      16:20:09.387
ClientBeginRequest:   16:20:09.393
GotRequestHeaders:    16:20:09.393
ClientDoneRequest:    16:20:09.393
Determine Gateway:    0ms
DNS Lookup:           0ms
TCP/IP Connect:       41ms
HTTPS Handshake:      0ms
ServerConnected:      16:20:09.435
FiddlerBeginRequest:  16:20:09.435
ServerGotRequest:     16:20:09.439
ServerBeginResponse:  16:20:09.465
GotResponseHeaders:   16:20:09.465
ServerDoneResponse:   16:20:09.465
ClientBeginResponse:  16:20:09.465
ClientDoneResponse:   16:20:09.466

      Overall Elapsed:     0:00:00.073

RESPONSE BYTES (by Content-Type)
--------------
~headers~: 208

ESTIMATED WORLDWIDE PERFORMANCE
```

Show Chart

工具面板
session
操作面板
统计、数据详情面板

ALT+Q > type HELP...

Capturing All Processes 1 / 65 http://i.firefoxchina.cn/newtab/images/upload/20120827191803_letv.png

火龙果·整理
uml.org.cn

Chapter 2 HTTPie接口测试工具

2.1 HTTPie介绍

2.2 HTTPie用法简介

2.1 HTTPie介绍

HTTPie （读aych-tee-tee-pie）是一个 HTTP 的命令行客户端，使用Python语言编写，常被用于手工接口测试。

特点

直观的语法
格式化和色彩化的终端输出
内置 JSON 支持
支持上传表单和文件
HTTPS、代理和认证
任意请求数据
自定义头部
持久性会话
类 Wget 下载
支持 Python 2.6, 2.7 和 3.x
支持 Linux, Mac OS X 和 Windows
插件
文档
测试覆盖率

安装：

首先安装python环境(下载，安装，设置环境变量）
然后安装pip install httpie
看到如下输出时，表示安装成功：

使用cmd 进入命令行界面：

```
C:\Users\zhiwang>http
usage: http-script.py [--json] [—form] [—pretty {all,colors,format,none}]
                      [--style STYLE] [—print WHAT] [—verbose] [--headers]
                      [--body] [—stream] [--output FILE] [—download]
                      [--continue]
                      [--session SESSION_NAME_OR_PATH | --session-read-only SESSION_NAME_OR_PATH]
                      [--auth USER[:PASS]] [—auth-type {basic,digest}]
                      [--proxy PROTOCOL:PROXY_URL] [—follow]
                      [--verify VERIFY] [—cert CERT] [—cert-key CERT_KEY]
                      [--timeout SECONDS] [--check-status] [--ignore-stdin]
                      [--help] [—version] [—traceback] [—debug]
                      [METHOD] URL [REQUEST_ITEM [REQUEST_ITEM ...]]
http-script.py: error: too few arguments
```

2.2 HTTPie用法简介


```
C:\Users\Administrator>http 192.168.103.100:5000/get
HTTP/1.0 200 OK
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: *
Content-Length: 279
Content-Type: application/json
Date: Sun, 30 Apr 2017 05:21:53 GMT
Server: Werkzeug/0.12.1 Python/2.6.6

{
    "args": {},
    "headers": {
        "Accept": "*/*",
        "Accept-Encoding": "gzip, deflate",
        "Connection": "keep-alive",
        "Host": "192.168.103.100:5000",
        "User-Agent": "HTTPie/0.9.9"
    },
    "origin": "192.168.103.1",
    "url": "http://192.168.103.100:5000/get"
}
```

Fiddler上的抓包：

Statistics | Inspectors | AutoResponder

Headers | TextView | SyntaxView | WebForms

Request Headers

GET /get HTTP/1.1

Client

Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: HTTPie/0.9.9

Transport

Connection: keep-alive
Host: 192.168.103.100:5000

Transformer | Headers | TextView | SyntaxView


```
{
  "args": {},
  "headers": {
    "Accept": "*/*",
    "Accept-Encoding": "gzip, deflate",
    "Connection": "keep-alive",
    "Host": "192.168.103.100:5000",
    "User-Agent": "HTTPie/0.9.9"
  },
  "origin": "192.168.103.1",
  "url": "http://192.168.103.100:5000/get"
}
```

2.2 HTTPie用法简介

练习以下命令,熟悉响应格式,状态码，并在Fiddler上详细察看抓包情况：

•HTTP GET方法：

http 192.168.103.102:5000/get

•修改Headers:

http 192.168.1.251:5000/get User-Agent:mtt/1.0 'Cookie:a=b;b=c'

•详细信息

http 192.168.1.251:5000/get -v

•只输出Headers：

http 192.168.1.251:5000/get -h

•HTTP POST方法 表单:

http --form POST 192.168.1.251:5000/post custname=hxy

•HTTP POST方法 数据：

http POST 192.168.1.251:5000/post custname=hxy

•HTTP PUT方法 数据：

http PUT 192.168.1.251:5000/put custname=hxy

•HTTP DELETE方法:

http DELETE 192.168.1.251:5000/delete

•基本认证：

http 192.168.1.251:5000/basic-auth/user/passwd -a user:passwd

火龙果·整理
uml.org.cn

Chapter 3 Postman接口测试工具

3.1 Postman介绍

3.2 Postman用法介绍

3.1 Postman介绍

Postman是google开发的一款功能强大的网页调试与发送网页HTTP请求工具，全世界有超过3万家公司及300万开发人员使用，其主要功能包括：

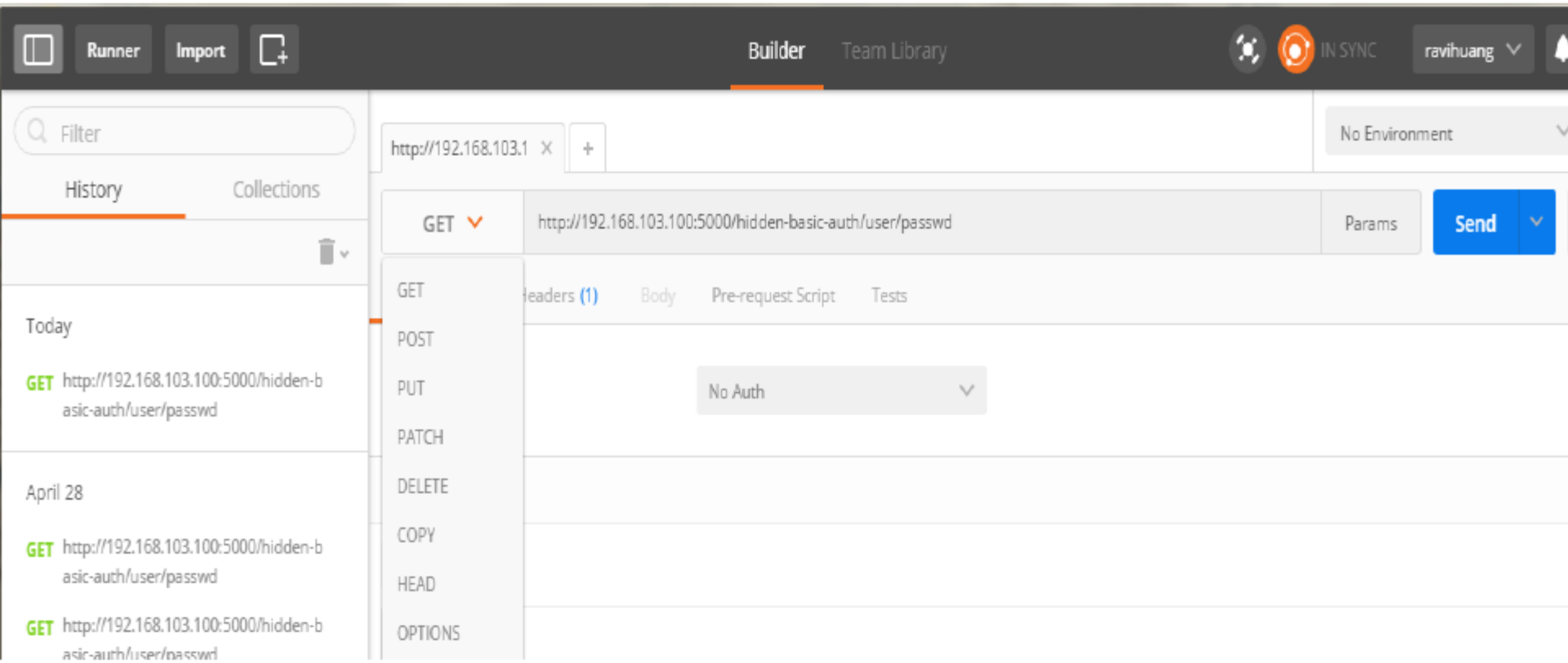
- 模拟各种HTTP requests
- Collection功能(测试集合):一个collection可以包含多条request
- 人性化的Response整理
- 内置测试脚本语言:Postman支持编写测试脚本，可以快速的检查request的结果
- 设定变量与环境

下载地址：https://www.getpostman.com/apps

3.2 Postman用法简介

火龙果·整理
uml.org.cn

Postman可以以客户端方式或者Chrome插件的方式使用，其主要功能与httpie类似，主界面如下图所示：



火龙果·整理
uml.org.cn

Chapter 4 Python自动化接口测试框架

4.1 什么是自动化测试框架

4.2 自动化测试框架组成

4.3 Python自动化测试框架

4.4 认识测试脚本

4.5 接口自动化测试框架

4.6 常见问题处理例程

4.1 什么是自动化测试框架

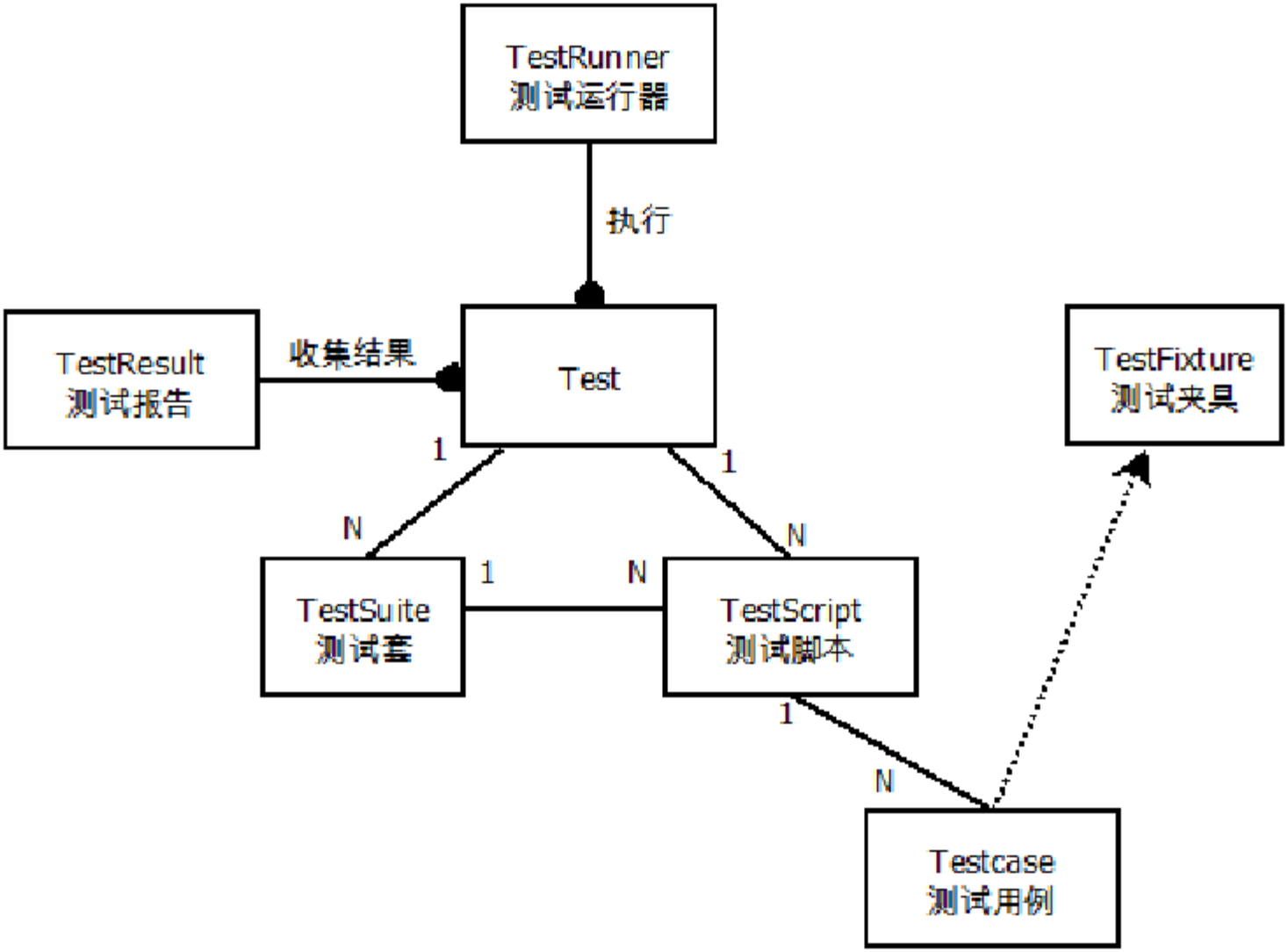
自动化测试框架：由一个或多个自动化测试基础模块、自动化测试管理模块、自动化测试统计模块等组成的工具集合。
按框架的定义来分，可以分为：基础功能测试框架、管理执行框架；
按不同的测试类型来分，可以分为：功能自动化测试框架、性能自动化测试框架；
按测试阶段来分，可以分为：单元自动化测试框架、接口自动化测试框架、系统自动化测试框架；
按组成结构来分，可以分为：单一自动化测试框架、综合自动化测试框架；
按部署方式来分，可以分为：单机自动化测试框架、分布式自动化测试框架。

常见自动化测试框架：
1，Selenium：通过浏览器进行的Web系统功能自动化测试工具
2，TestNG/Junit：Java单元测试框架
3，QTP：Windows客户端软件功能自动化测试工具
4，RF：RobotFramework，集成/系统通用自动化测试工具

常见自动化接口测试框架：
1，基于Java语言：
TestNG+HttpClient+JDBC
2，基于Python语言：
Pytest+Requests+Pymysql
3，RF

4.2 自动化测试框架组成

火龙果·整理
uml.org.cn



4.3 Python测试框架(安装)

Python语言常见测试框架有unittest和pytest，前者是自带的，后者需要安装这里介绍pytest

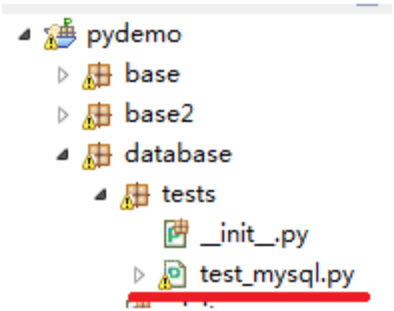
安装

```
C:\Users\Administrator>pip install pytest
Collecting pytest
  Using cached pytest-3.0.7-py2.py3-none-any.whl
Requirement already satisfied (use --upgrade to u
\python27\lib\site-packages (from pytest)
Requirement already satisfied (use --upgrade to u
\python27\lib\site-packages (from pytest)
Requirement already satisfied (use --upgrade to u
ython27\lib\site-packages (from pytest)
Installing collected packages: pytest
Successfully installed pytest-3.0.7
```

确认安装成功

```
C:\Users\Administrator>pytest --version
This is pytest version 3.0.7, imported f
```

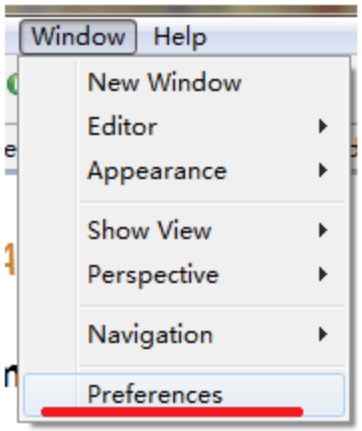
创建一个测试用例及脚本：

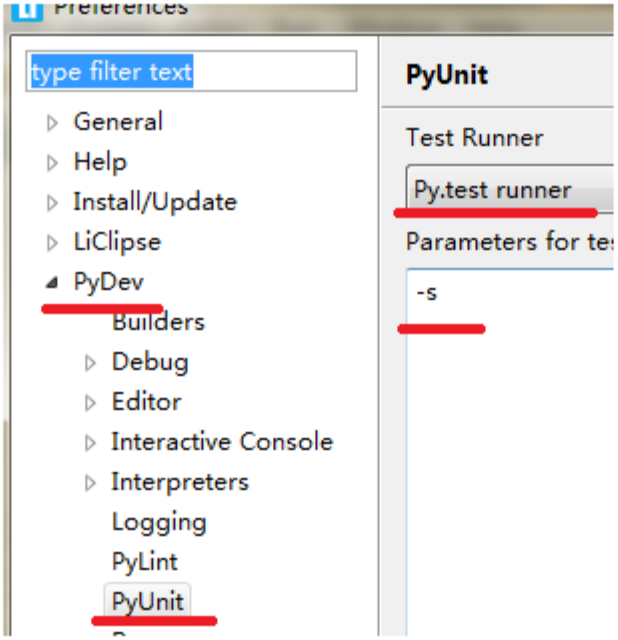


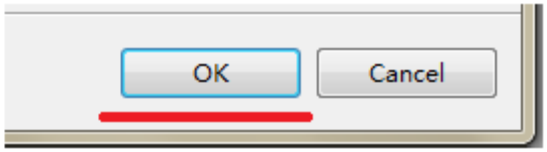

```
from database import demo_mysql as mysql
import pytest

def test_mysql_select():
    data=mysql.select_one_from("EMP")
    assert len(data)==8
```

4.3 Python测试框架（配置）

配置TestRunner：
在Eclipse/Python上配置python的TestRunner:
1，在windows菜单上选择Preference
2，在Preferences->PyDev->PyUnit中选择Py.test runner
3，配置-s参数
4，OK按钮结束配置

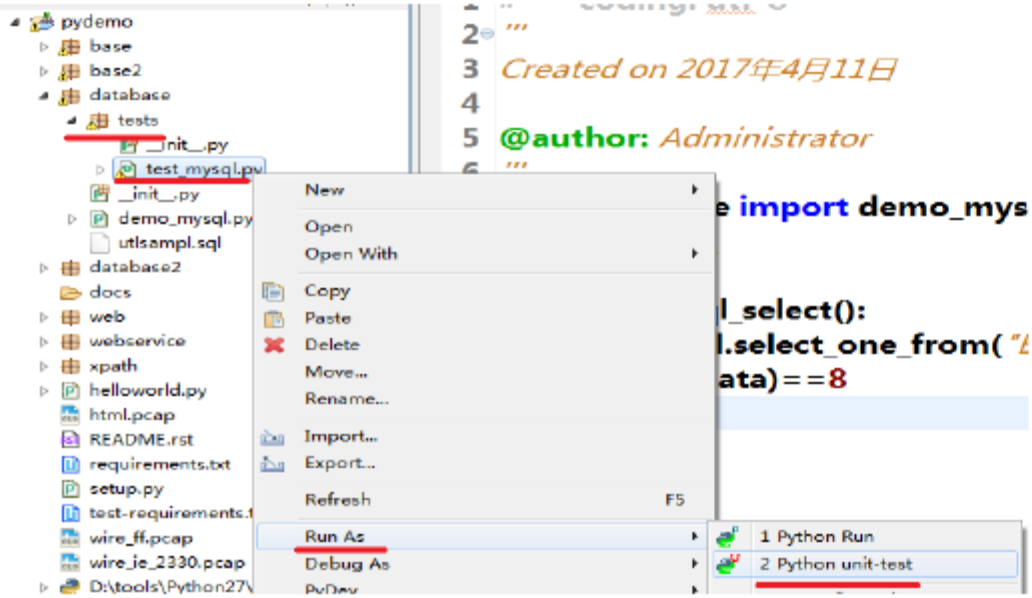




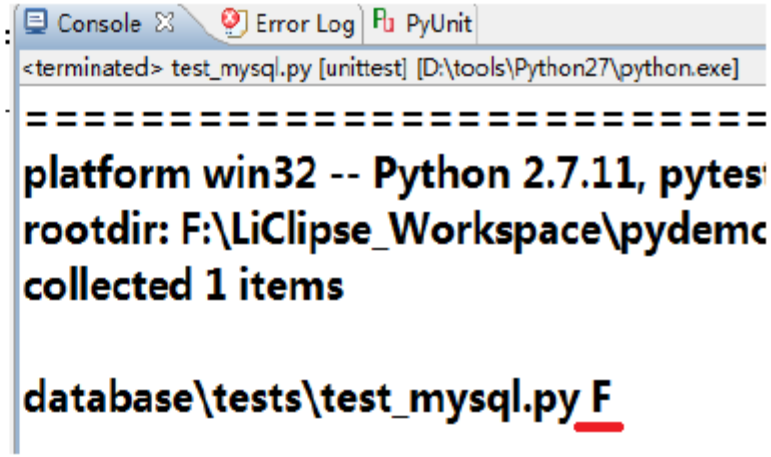


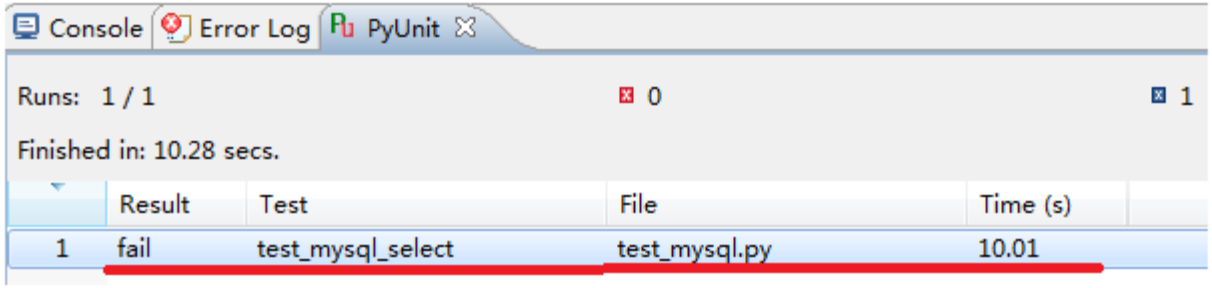
4.3 Python测试框架(执行)

在工程里如下依次选择Python unit-test菜单项，点击即可执行该测试脚本：



Console中输出的F就是测试结果





4.4 认识测试脚本模式


```
from database import demo_mysql as mysql    # 待测模块
import pytest                               # 测试框架

def test_mysql_select():                    # 测试方法,必须以test开头,
                                            # 可以没有下划线
    data=mysql.select_one_from("EMP")
    assert len(data)==8                     # 断言，理解断言
```

自动化测试就是将手工测试的步骤通过脚本语言来控制计算机，让计算机自动执行测试方法。自动化测试脚本是一系列**测试操作自动化**和**检查点校验自动化**的组合；

assert中文意思为断言，后面跟一个条件表达式，当表达式求值结果为True时断言成功，否则测试用例执行失败。

断言是自动化测试中实现结果检查自动化的方法

4.4 认识测试脚本(TestFixture)

隐含fixture:

```
def setup_module(module):
    print ("\n<setup_module>")

def teardown_module(module):
    print ("</teardown_module>")

def setup_function(function):
    print ("<setup_function>")

def teardown_function(function):
    print ("</teardown_function>")
```

TestFixture 指的是测试中依赖的环境数据和条件等。通过TestFixture,测试框架可以更好的管理、重用测试环境,测试数据及测试脚本，以提高测试套的稳定性和执行速度。

Python装饰器实现fixture:

```
#2， autouse的fixture
@pytest.fixture(scope="session",autouse=True)
def setup_session(request):
    print('\n<session>')
    def teardon_session():
        print('</session>')
    request.addfinalizer(teardon_session)

@pytest.fixture(scope="module",autouse=True)
def setup_module(request):
    print('<module>')
    def teardown_module():
        print('</module>')
    request.addfinalizer(teardown_module)

@pytest.fixture(scope="function",autouse=True)
def setup_function(request):
    print("<function>,%s'"%(request))
    def teardown_function():
        print('\n</function>')
    request.addfinalizer(teardown_function)
```

4.5 接口自动化测试框架(工程)

PyDev Package Explorer

- ifTest
  - libs
    - __init__.py
    - excel.py
    - mysql.py
  - tests
    - __init__.py
    - BasicTestCase.py
    - conftest.py
    - test_01_basic.py
    - test_02_auth.py
    - test_02_class.py
    - test_03_post.py
    - test_04_put.py
    - test_05_cache.py
    - test_06_headers.py
    - test_07_query.py
    - test_08_cookie.py
    - test_09_xml.py
    - test_10_excel.py
  - ~$testcases.xlsx
  - log.txt
  - README.rst
  - report.html
  - requirements.txt
  - testcases.xlsx
  - D:\tools\Python27\python.exe

公共库

测试套

测试脚本

Excel格式的测试用例

自动化测试框架通常都是类似左图的工程结构

4.5 Python接口自动化测试框架

这里的Python HTTP接口自动化测试框架采用python的以下开发库实现：

pytest: 提供TestRunner， TestFixture， TestResults功能，用来管理和组织测试脚本

pymysql： 为测试脚本提供访问mysql数据库的能力

requests： 为测试脚本提供发送HTTP请求，接收响应，以及处理用户数据的能力

cx_oracle： (可选)：为测试脚本提供访问oracle数据库的能力

xlrd(可选)： 为测试脚本提供处理Excel文件的功能

lxml(可选)： 为测试脚本提供处理xml类型用户数据的功能

requirements.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
pytest
pymysql
requests
lxml
cx_oracle
```

```
F:\LiClipse_Workspace\ifTest>pip install -r requirements.txt
```

安装方式

4.5 接口自动化测试框架(用例)

火龙果·整理
uml.org.cn

```python
#!/usr/bin/python
# -*- coding: utf-8 -*-
'''
Created on 2017年4月28日

@author: 黄小勇
'''
import pytest
from conftest import cfg

url=cfg['BASE_URL']+"basic-auth/user/passwd"

# 普通测试用例
def test_basic_auth():
    print "%s http 基本认证-合法的用户名和密码: " % cfg['PREFIX']
    response = cfg['S'].get(url, auth=("user", "passwd"))
    assert response.status_code == 200

# 参数化测试用例
data_401 = [
    ("baduser","passwd"),
    ("user","badpasswd"),
]
@pytest.mark.parametrize("uname,passwd", data_401)
def test_basic_auth_401(uname,passwd):
    print "%s http 基本认证: 用户名或者密码错误: " % cfg['PREFIX']
    response = cfg['S'].get(url, auth=(uname, passwd))
    assert response.status_code == 401
```

测试用例的参数化使用Python装饰器技术实现

4.6 常见问题处理例程

火龙果·整理
uml.org.cn

BasicTestCase.py	2017/4/30 20:09	Python File	1 KB	
conftest.py	2017/5/2 13:23	Python File	2 KB	TestFixture
test_01_basic.py	2017/5/2 8:53	Python File	1 KB	
test_02_auth.py	2017/4/30 19:55	Python File	1 KB	HTTP鉴权demo
test_02_class.py	2017/4/30 19:55	Python File	1 KB	使用类来组织测试用例
test_03_post.py	2017/4/29 10:35	Python File	1 KB	
test_04_put.py	2017/4/29 10:35	Python File	1 KB	
test_05_cache.py	2017/4/29 10:35	Python File	1 KB	HTTP Cache类Demo
test_06_headers.py	2017/4/29 10:50	Python File	1 KB	
test_07_query.py	2017/4/29 11:14	Python File	1 KB	
test_08_cookie.py	2017/4/29 11:37	Python File	1 KB	HTTP Cookie Demo
test_09_xml.py	2017/4/30 19:55	Python File	1 KB	处理XML类型用户数据
test_10_excel.py	2017/4/30 19:55	Python File	2 KB	通过Excel读取测试用例