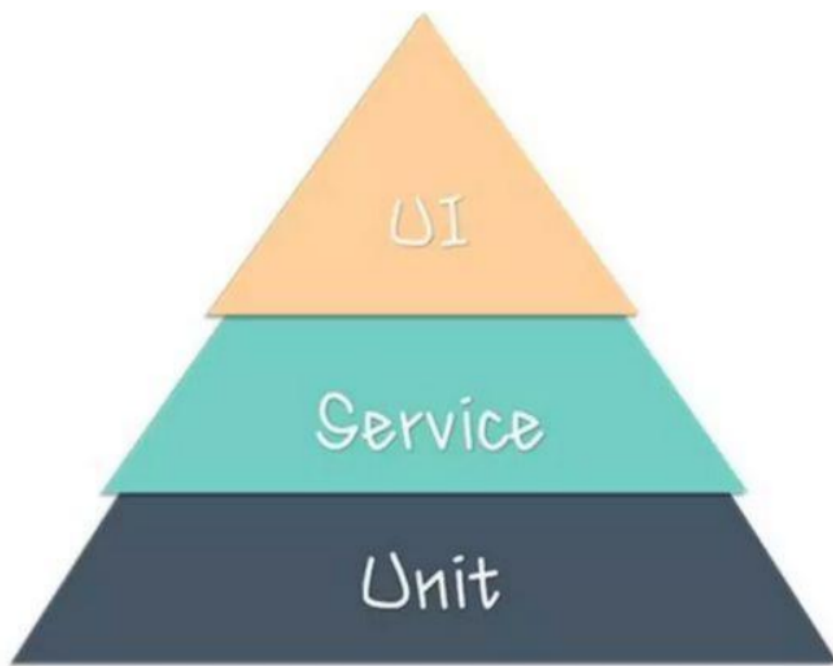


## 背景介绍

在测试金字塔模型中分为三层：最底层是单元测试，中间层接口测试，最顶层为 UI 测试。接口测试位于第二层。接口一般比较稳定，因此接口测试的效益最大,同时接口测试可以跨应用：Web/App 跨平台：Win, Mac,Liunx等



[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

## 什么是接口

API (Application Programming Interface,应用程序编程接口) 是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。

举例说明：

- 笔记本电脑上提供了各种物理硬件接口，比如 USB 接口，耳机接口、麦克风接口、电源接口等.... 这些不同的接口有不同的功能：比如通过 USB 接口插入 U 盘就可以拷贝电脑数据，插入耳机接口可以听音乐，我们无需关心这些接口的工作原理，只需通过这些接口满足我们的使用需求即可。
- 在中国天气网网上查询某个城市天气，主要输入城市名称，即可获取到对应的天气。查询背后的本质也是调用了网站后台接口来获取数据，这里的接口是 Web 服务软件接口。用户不需要关注数据在网站后台是怎么查询的，只需要返回一个结果即可。

## 接口测试

### 什么是接口测试？

接口测试是对系统或组件之间的接口进行测试，主要是校验数据的交换，传递和控制管理过程，以及相互逻辑依赖关系。其中接口协议分为 HTTP ,WebService ,Dubbo ,Thrift ,Socket 等类型。测试类型又主要分为功能测试，性能测试，稳定性测试，安全性测试等。

## 为什么要做接口测试？

接口测试实施在多系统的平台架构下，有着极为高效的成本收益比（当然，单元测试收益更高，但实施单元测试的成本投入更大，技术要求更高）。

接口测试天生为高复杂性的平台带来高效的缺陷检测和质量监督能力，平台复杂，系统越庞大，接口测试的效果越明显。

接口测试优势主要体现在如下三个方面：

### 1、节省了测试成本

根据数据模型推算，底层的一个程序 BUG 可能引发 UI 层的 8 个左右 BUG，而且底层的 BUG 更容易引起全网的死机；接口测试能够提供系统复杂度上升情况下的低成本高效率的解决方案。

### 2、接口测试门槛相对较低

接口测试不同于单元测试，接口测试是站在用户的角度对系统接口进行全面高效持续的检测。

### 3、效益更高

将接口测试实现为自动化和持续集成，当系统复杂度和体积越大，接口测试的成本就越低，相对应的，效益产出就越高。

## 怎么样去做接口测试？

基本的接口功能自动化测试流程如下：

需求分析 -> 用例设计 -> 脚本开发 -> 测试执行 -> 结果分析

- 需求分析  
需求分析是参考需求、设计等文档，在了解需求的基础上还需清楚内部的实现逻辑，并且可以在这一阶段提出需求、设计存在的不合理或遗漏之处。
- 用例设计  
用例设计是在理解接口测试需求的基础上，使用 Excel 或 XMind 等思维导图软件编写测试用例设计，主要内容包括参数校验，功能校验、业务场景校验、安全性及性能校验等，常用的用例设计方法有等价类划分法，边界值分析法，场景分析法，因果图，正交表等。
- 脚本开发  
使用自己熟悉的语言或者工具来开发即可。
- 测试执行  
可以直接运行接口测试脚本或者使用 CI 来执行测试
- 结果分析  
根据测试报告来分析接口测试结果。

## json数据

### JSON 简介

JSON (JavaScript Object Notation, 即 JavaScript 对象表示法) 是一种轻量级的数据交换格式。它独立于语言和平台，JSON 解析器和 JSON 库支持不同的编程语言。JSON 具有自我描述性，很容易理解。目前大多数接口返回

的数据格式为 JSON,因此进行接口测试必须掌握 JSON。

### JSON 语法

语法规则

- 数据在键/值对中
- 数据由逗号分隔
- {花括号}保存对象
- [方括号]保存数组

## JSON 键/值对

JSON 数据的书写格式是：key:value 键值对。比如：

```
1 | "Name" : "51zxw"
```

JSON 值可以是：

- 数字（整数或浮点数）
- 字符串（在双引号中）
- 逻辑值（true 或 false）
- 数组（在方括号中）
- 对象（在花括号中）
- null

## JSON 数字

```
1 | "status_code":200
```

## JSON 字符串

```
1 | "Name" : "51zxw"
```

## JSON 逻辑值

```
1 | "result":true
```

## JSON 数组

```
1 | "user":["51zxw","zxw2018","zxw666"]
```

## JSON 对象

JSON 对象在花括号中书写：对象可以包含多个键/值对：

```
1 | { "firstName":"John" , "lastName":"Doe" }
```

Tips:在接口测试过程中，一般都是返回 JSON 对象类型。

## JSON 数据嵌套

比如在数组中含多个对象：

```
1 | {  
2 |   "employees": [  
3 |     { "firstName":"John" , "lastName":"Doe" },  
4 |     { "firstName":"Anna" , "lastName":"Smith" },  
5 |     { "firstName":"Peter" , "lastName":"Jones" }  
6 |   ]  
7 | }
```

在上面的例子中，对象 "employees" 是包含三个对象的数组。每个对象代表一条关于某人（有姓和名）的记录。

## JSON 数据解析

Python3 中可以使用 json 模块来对 JSON 数据进行编解码，它包含了两个方法：

- json.dumps(): 将 python 数据转化为 Json 数据
- json.loads(): 将 json 数据类型转为 Python 数据类型

JSON 库官方文档 <https://docs.python.org/3/library/json.html>

```
1  # json.dumps()
2  # 将 python 数据转化为 Json 数据 json_dumps.py
3  import json
4  data={'id':1, 'name': '51zxw', 'password': '66666'}
5  print(type(data))
6  json_str=json.dumps(data)
7  print(type(json_str))
8  print(json_str)
9  # 输出结果:
10 <class 'dict'>
11 <class 'str'>
12 {"password": "66666", "id": 1, "name": "51zxw"}
13
14 # json.loads()
15 # 将 json 数据类型转为 Python 数据类型 json_loads.py
16 import json
17 json_str='{"id":1,"name":"51zxw","password":"66666"}'
18 data=json.loads(json_str)
19 print(type(json_str))
20 print(type(data))
21 print(data)
22 print(data['id'])
23 print(data['name'])
24 # 输出结果:
25 <class 'str'>
26 <class 'dict'>
27 {'name': '51zxw', 'password': '66666', 'id': 1}
28 1
29 51zxw
```

## Json 文件处理

有时我们可能需要将 JSON 数据写入到文件，或者从 Json 数据文件读取数据

## 写入 JSON 数据到文件

```
1  with open('data.json', 'w') as f:
2      json.dump(data, f)
```

## 读取 JSON 数据文件

```
1  with open('data.json', 'r') as f:
2      data = json.load(f)
```

## 2、接口测试:接口测试（重点）

### A、是什么？

测试后端实现是否符合接口规范

### B、为什么？

程序安全、效率

### C、怎么用？(实现流程)

模拟客户向服务器发送数据，然后查看响应

要素1: 定位服务器资源(通过 URL 实现: <http://www.baidu.com> | <http://127.0.0.1:8000/资源路径>)

要素2: 模拟用户提交数据

要素3: 查看响应的结果是否符合预期

### D、接口测试分类(记住)

类型1: web接口测试 (BS 架构)

| ---- 服务器接口测试: 测试自己公司实现的接口

| ---- 第三方接口测试: 测试别的公司实现的接口

类型2: 模块接口测试 (CS 架构)

[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

## 3、接口测试:环境搭建

### A、是什么？

搭建接口测试的测试环境，运行程序

### B、为什么？

必须的

### C、怎么用？

1)、运维或开发实现程序部署，正常运行

2)、自己搭建

2-1、安装 python

2-2、安装依赖 : `pip install -r requirements.txt -i https://pypi.douban.com/simple`

2-3、启动项目 : `python run_server.py`

2-4、测试: <http://127.0.0.1:8000/api/departments> 如果正常响应数据了, 那么程序安装并启动正常

### C、怎么用？

1)、运维或开发实现程序部署，正常运行

2)、自己搭建

2-1、安装 python

2-2、安装依赖 :pip install -r requirements.txt -i <https://pypi.douban.com/simple>

2-3、启动项目 : python run\_server.py

2-4、测试: <http://127.0.0.1:8000/api/departments> 如果正常响应数据了，那么程序安装并启动正常

3)、学生管理系统简介

核心: 学生表 班级表 学院表

操作: 3张表的增删改查

[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

## 4、接口测试:插件

### A、是什么？

是测试接口的测试工具

火狐 ----- RESTClient

谷歌 ----- POSTman

Java 实现的测试工具: Jmeter(重点)

### B、为什么？

测试工具功能强大一些，使测试更快捷方便

### C、怎么用？

I

[火狐安装 RESTClient](#)

[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

## 5、接口测试:RESTful 风格(重点)

### A、是什么？

**RESTful 风格：**开发测试时，大家都会遵守一些默认规范，这些是约定成俗的，大家的习惯做法，非强制性的。RESTful 风格就是这些习惯中的一种。

RESTful 风格中规范了，URL 的实现方式，提交的数据的实现方式，响应的数据的实现方式 .....

### B、为什么？

为了程序的易读性和易维护性

[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

### C、怎么用？

RESTful 风格是约束接口操作的四种操作以及三个要素

增:

要素1: URL + POST

要素2: 多个数据

方式1: 键=值&键=值

方式2: JSON 格式

要素3: 200 | 201 + 添加的记录

[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

查:

要素1: URL + GET

要素2:常用方式1 URL?键=值&键=值....

要素3:200 + 一条记录或多条记录



改:

要素1:URL + POST | PUT(更合适)

要素2:

方式1: 键=值&键=值

方式2: JSON 格式s

要素3:200 | 201 + 修改后那条记录

[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

删:

要素1: URL + GET | DELETE

要素2: 方式1: 键=值&键=值

要素3: 200 | 204 + 无

状态码:

404 ----- 资源路径有误

505 ----- 服务器异常

## 6、接口测试:JSON

### A、是什么?

A-1、JSON 是一种数据载体

A-2、互联网本质就是数据传输, 数据传输需要数据载体,比如: 页面信息就是存储在 html 这种数据载体中的

A-3、HTML 格式

```
<html>
  <head>
    <title>hello</title>
  </head>
  <body>
    <font color='red'> hello world </font>
  </body>
</html>
```

[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

xml 格式:

```
<person>
  <name>huluwa</name>
  <age>8</age>
</person>
```



缺点:

标签标记语言, 有效数据占有率低

#### A-4、JSON 优化数据传输

```
{"title":"hello","font":"hello world"}
```

```
{"name":"huluwa","age":"8"}
```

#### B、为什么?

JSON 传输数据效率更高, 所以部分场景下使用 JSON 替换 html 和 XML

但是 JSON 语法描述性不及 标签语言, 所以部分场景还得使用 html 和 xml

如果传递的是少量数据的话, 可能使用 JSON

#### C、怎么用?

语法:

格式1(JSON对象):{"键1":"值1","键2":"值2".....}

格式2(JSON数组):[值1,值2,值3.....]

格式复合: {"name":"huluwa","age":"8","aihao":["救爷爷","吐火","吐水"]}

[{"name":"huluwa","age":"8"}, {"name":"aotuman","age":"10"}.....]

[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

## 1、Jmeter: 概述

#### A、是什么?

Jmeter 是 Apache 公司使用 Java 开发的一款测试工具

#### B、为什么?

高效、功能强大

模拟一些高并发或多次循环等特殊测试场景

#### C、怎么用?

C-1、下载 Jmeter, 解压缩

C-2、安装 Java 环境(Jmeter 和 Java 环境的关系, 类似于mp3 文件和 播放器关系)

C-3、简单了解 Jmeter 的目录结构

C-4、启动 Jmeter

C-5、编写一个 Jmeter 版本的测试案例

#### D、Jmeter 的作用优点以及缺点(了解)

作用:

1. 接口测试
2. 性能测试 : 内在 ----- 程序的算法
3. 压力测试 : 外在 ----- 外在负载
4. Web自动化测试
5. 数据库测试 : 测试数据库
6. JAVA程序测试

优点:

1. 开源、免费
2. 支持多协议 : http https ftp ftps .....
3. 小巧
4. 功能强大



## unittest与requests结合

案例：tpshop登陆 用例数3

结构：

- setUp:
  - 作用：test开始方法执行之前，首先会被执行。
- tearDown:
  - 作用：test开始方法执行之后，会被执行。
- test\_login\_success:
  - 作用：登陆成功用例
- test\_username\_not\_exist:
  - 作用：登陆失败，账号不存在
- test\_password\_error:
  - 作用：登陆失败，密码错误

[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

注意：

- 测试类必须继承unittest.TestCase
- 测试用例方法，必须以test开头

```
def test_login_success(self):
    # 请求验证码 --> 获取cookies
    self.session.get(self.url_verify)
    # 请求登陆
    data = {"username": "13800001111",
           "password": "123456",
           "verify_code": 8888}
    r = self.session.post(self.url_login, data=data)
    try:
        # 断言
        self.assertEqual("登陆成功", r.json()['msg'])
    except AssertionError as e:
        print(e)
```

[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

案例：tpshop登陆 用例数3

结构：

- setUp:
  - 作用：test开始方法执行之前，首先会被执行。
  - 应用：
    - 获取session对象
    - 登陆url
    - 验证码url
- tearDown:
  - 作用：test开始方法执行之后，会被执行。
  - 应用：
    - 关闭session对象

[https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

- test\_login\_success:
  - 作用：登陆成功用例
  - 应用：
    - 请求验证码—>让session获取并记录cookies信息
    - 请求登陆
    - 断言

## 2、什么是unittest,自动化测试中unittest能自动化测试解决什么问题（意义是什么）

unittest是python内置模块，其实也是自动化测试的自动化测试框架

该模块提供很多类和方法来处理测试各种工作

unittest的四大组件？

- 1) testcase
- 2) testfixture
- 3) testsuite
- 4) testrunner 测试运行器

UnitTest框架：

Python自带的一个作为单元测试的测试框，PyUnit，相当于Java中的JUnit，随着自动化技术的成熟，UnitTest成为了测试框架第一选择，可以完整的结合Selenium、Requests来实现UI和接口的自动化，由UnitTest再衍生出PyTest，PyTest可以完美结合UnitTest来实现自动化。

鄙视链：PyTest——UnitTest——RobotFrameWork

基本应用：

1. 环境搭建，Python中已经直接加载了UnitTest框架，无须额外安装

2. 四大组件：

a. test fixture：setUp（前置条件）、tearDown（后置条件），用于初始化测试用例及清理和释放资源

b. test case：测试用例，通过集成unittest.TestCase，来实现用例的继承，在UnitTest中，测试用例都是通过test来识别的。

c. test suite：测试套件，也称之为测试用例集

d. test runner：运行器，一般通过runner来调用suite去执行测试 [https://blog.csdn.net/z\\_202041](https://blog.csdn.net/z_202041)

我们还需要产生测试报表，所以要安装一个第三方插件 **pytest-html**，执行如下命令安装

```
pip install pytest-html
```