



蜗牛学院

重构IT职业教育新生态

接口测试

目录

C O N T E N T S

1

接口测试简介

2

接口测试工具

3

接口测试设计

4

接口测试实践

基础知识

1. 接口及接口测试的定义
2. 深入了解HTTP协议 (Post, Put, Delete, Get, URL构成, 头部, 正文, 状态码)
3. 熟练使用抓包工具(Fiddler, Httpwatch, Wireshark)
4. 熟悉Cookie和Session
5. 熟悉测试流程(分析-设计-实现-执行)

目录

C O N T E N T S

1

接口测试简介

2

接口测试工具

3

接口测试设计

4

接口测试实践

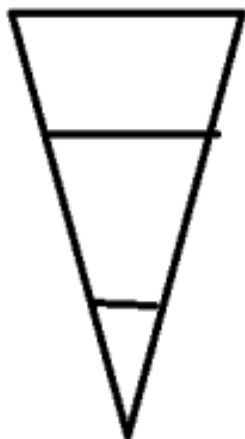
接口测试的重要性

相对于UI测试，能更早的发现问题

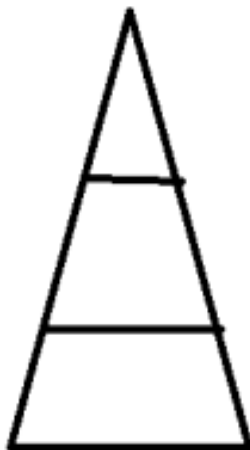
相对于UI测试，能发现更底层的问题

相对于UI测试，解决的效率更高

相对于UI测试，稳定性更高



测试冰淇淋



测试金字塔



测试橄榄球

接口测试-测试要点

主要测试要点如下：

- 检查接口的功能, 接口返回的数据是否与预期结果一致
- 检查接口的容错性和健壮性, 分析接口传参的边界
- 检查接口的性能, 对于一些接口, 需要进行多线程测试 (适用于接口压力测试)
- 接口的安全性

目录

C O N T E N T S

1

接口测试简介

2

接口测试工具

3

接口测试设计

4

接口测试实践

测试工具

1. Poster
2. Fiddler
3. Postman
4. LoadRunner
5. Jmeter
6. SoapUI
7. insomnia(Postman弱化版, 开源)
8. paw(mac上最强)

目录

C O N T E N T S

1

接口测试简介

2

接口测试工具

3

接口测试设计

4

接口测试实践

接口测试设计

1. 分层思想

- Testcase与接口业务脚本分离

2. 好处

- 提高业务脚本复用性
- 方便维护
- 通用的工具便于不同项目调用和封装

目录

C O N T E N T S

1

接口测试简介

2

接口测试工具

3

接口测试设计

4

接口测试实践

实现接口测试

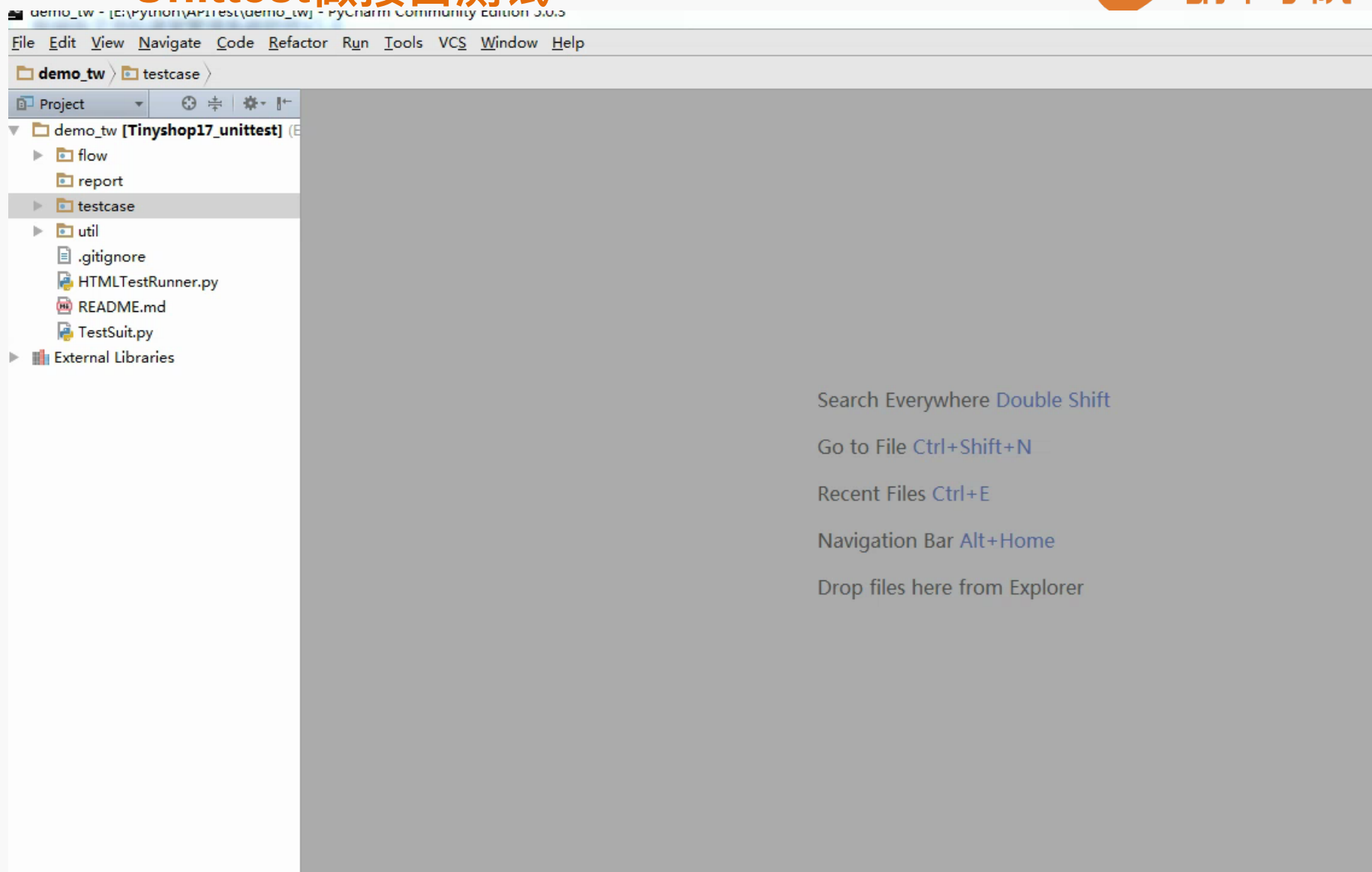
一. unittest框架

1. setUp
2. TestCase
3. tearDown

二. 编写代码

三. 生成并分析测试报告

Unittest做接口测试



接口测试最佳实践

1. 加密接口，数字签名
2. HTTPS
3. Ajax请求
4. 接口测试数据的准备
5. 业务逻辑复杂，接口之间进行数据的传递
6. 接口的依赖，外部接口调用
7. 接口的安全性
8. 接口的性能
9. 接口文档的管理
10. 持续集成
11. Web前端性能测试

1. 带数字签名的接口

名称	添加发布会接口
描述	用来添加发布会的接口
URL	http://127.0.0.1:8000/api/sec_add_event/
调用方式	POST
传入参数	<pre> eid #发布会id int name #发布会标题 string limit # 限制人数 int address # 地址 string start_time # 发布会时间 'YYYY-MM-DD HH:MM:SS' time #当前时间 时间戳前10位 比如: 1472052712.9126902 sign #签名 </pre>
备注	<pre> sign 计算公式: Md5(time+api_key) 其中api_key='&Guest-Bugmaster' </pre>

2. HTTPS

1. 受信任主机

- 代码与发送HTTP请求完全一样，没有任何区别

2. 不受信任主机

- 增加了context参数并禁用验证即可

```
import http.client, ssl

# 禁用所有的验证
context = ssl._create_unverified_context()
# 与服务器建立连接, 并使用context参数
conn = http.client.HTTPSConnection('localhost', 8443, context=context)
url = '/agileone/'
# 向服务器发起GET请求
conn.request('GET', url)
# 获取响应并解码
response = conn.getresponse().read().decode()
# 打印响应
print(response)
```


3. Ajax请求

1. 添加头部字段

X-Requested-With: XMLHttpRequest

4. 接口测试数据的准备

1. 做数据一般不建议自己直接通过DB写数据，因为一般业务系统都比较复杂，直接写DB可能会造成某些关联表没有数据，导致测试数据不可用。
2. 某些业务数据可能会通过某种机制，比如MQ消息等通知到其他系统，其他系统接收到消息也会有业务处理，会导致直接写DB的业务数据不可用。

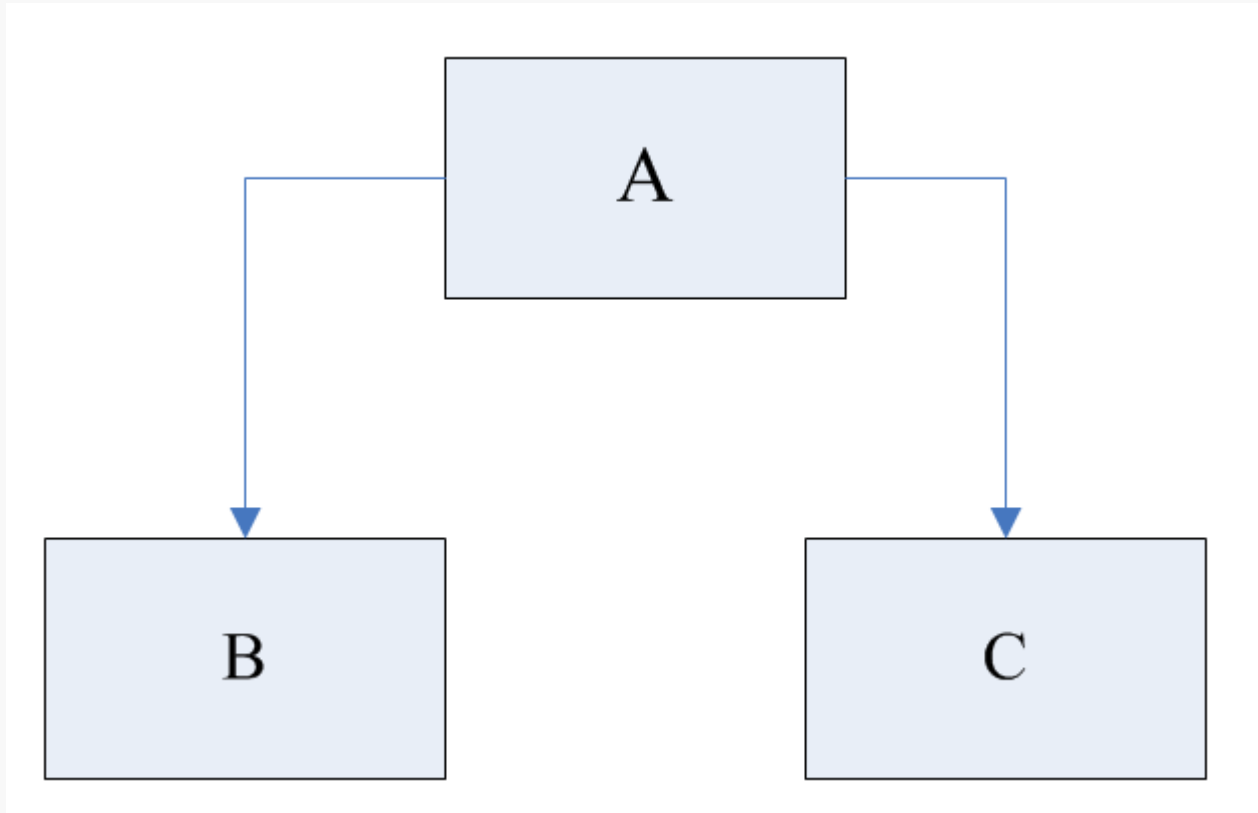
所以建议还是通过业务接口做数据。

5. 业务逻辑复杂，接口之间进行数据的传递



1. 那就传呗~

6. 接口的依赖，外部接口调用



1. python模块（mock, httmock, responses）
2. Java的WireMock
3. mock server（可以自己搭建，在线Mock server系统（<https://easy-mock.com>））

7. 接口的安全性测试

1. 敏感信息是否加密
2. SQL注入(1 or 1 = 1)
3. XSS跨站脚本攻击
4. XML注入
5. 文件上传漏洞(copy test.jpg/b+test.php newtest.jpg)
6. 前端和后台的验证是否一致

8. 接口的性能测试

实现:


1. Loadrunner
2. Jmeter
3. Locust







指标:

1. 并发用户数
2. 事务响应时间
3. 每秒事务数
4. 吞吐量
5. 服务器系统资源使用率(CPU, 内存, 磁盘, IO等)

9. 接口文档的管理

1. swagger(<http://editor.swagger.io/>)

pet Everything about your Pets Find out more: <http://swagger.io> 

POST	<code>/pet</code> Add a new pet to the store	
PUT	<code>/pet</code> Update an existing pet	
GET	<code>/pet/findByStatus</code> Finds Pets by status	
GET	<code>/pet/findByTags</code> Finds Pets by tags	
GET	<code>/pet/{petId}</code> Find pet by ID	
POST	<code>/pet/{petId}</code> Updates a pet in the store with form data	

POST	<code>/user/complex/add.htm</code>	新增用户(自定义参数)
GET	<code>/user/complex/{userId}/info.htm</code>	获取用户详情(多层嵌套)
GET	<code>/user/complex/list.htm</code>	查询用户列表(多层嵌套)
GET	<code>/user/complex/{userId}/info.htm?type=mock</code>	获取用户详情(支持mock)
POST	开发中 <code>/user/simple/{userId}/upload.htm</code>	上传图片

10. 持续集成

1. 接口用例编写调试完成后，部署到持续集成的测试环境中
2. 设定脚本运行频率，触发条件，告警方式等基本参数，进行接口的日常监控
3. 使用邮件发送测试结果

11. Web前端请求的测试分析

1. 传输压缩(Accept-Encoding: gzip, deflate)
2. 请求数优化
3. 时间段分析
4. 代码压缩

工具:

1. HTTPWatch
2. dynaTrace Ajax
3. YSlow

总结

接口测试, 想说爱你不容易, 功夫在诗外:

1. 业务逻辑
2. 协议(深刻理解)
3. 接口测试工具
4. 数据库基本操作
5. 编程语言
6. 前端
7. 安全常识

资料分享



1. HTTP下午茶

<http://www.kancloud.cn/kancloud/tealeaf-http/43836>

2. Python+Locust性能测试

<http://www.woniuxy.com/live/6>

3. Python接口测试自动化

<http://www.woniuxy.com/course/67>

4. Java接口测试自动化

<http://www.woniuxy.com/course/46>



谢谢聆听

蜗牛学院，只为成就更好的你！

