

基于UnitTest + PBT的接口测试用例批量生成

原创 三三三木 测试开发研习社 2022-05-19 10:55 发表于湖南

收录于合集

#自动化测试框架 7 #接口自动化测试 8 #requests 3

(全文约1000字，阅读约需3分钟，首发于公众号：测试开发研习社，欢迎关注)



测试开发研习社

分享测试开发的探索与实践，利用Python建设微服务时代下的持续测试方案

29篇原创内容

公众号

之前在公众号里已经介绍过 PBT 了，它可以为单元测试和接口测试自动、批量的生成用例。今天分享一下我使用 PBT 在接口测试中批量生成的封装示例，欢迎各位提出宝贵意见。

0. 封装思路

有接口测试经验的同学应该会发现，绝大部分的接口项目，测试步骤如下：

1. 阅读接口文档
2. 选择接口测试工具
3. 设计测试用例
4. 使用测试工具进行接口调用
5. 判断用例结果、输出测试报告

目前来看：

- 第 1、2、3 步 完全由技术人员手动完成
- 第 4 步可以通过代码完成，但代码要由技术人员写出
- 第 5、6 步通过测试框架（比如 unittest）完成

我的想法是 1-5 全部交给 Python 去完成，得益于 Python 庞大的生态，这个想法非常具有可行性：

- `apistar` 可以承担第 1、2、4 步
- `hypothesis` 可以承担第 3 步
- `unittest` 继续承担第 5、6 步

如此一来，除了一些特殊用例需要在第三步的时候由人工进行修改、补充外，
接口测试就可以**全部由 Python 自动完成了**，

1. 读懂接口

OpenAPI 接口文档示例

登陆

POST http://127.0.0.1:7600/login/access_token

Request

Body

```
password string<password>  
  >= 8 characters <= 20 characters  
email string<email>  
  >= 8 characters <= 30 characters
```

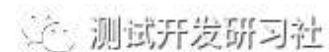
Responses

200 400 401 422

Successful Response

Body

```
access_token string  
token_type string
```



apistar 是一个 OpenAPI 工具，可以解析 OpenAPI (swagger) 的内容，其中包含了：

- 接口地址
- 接口请求方法
- 鉴权要求
- 接口参数
 - 参数名
 - 参数值类型

这样一来，我们就可以借助 apistar 让 Python“ 读懂 ”接口文档


apistar 代码示例

```
import apistar
import requests
from typesystem import to_json_schema

# 获取OpenAPI接口文档
openapi = requests.get("http://127.0.0.1:7600/openapi.json").text

client = apistar.Client(openapi) # 解析接口文档

link = client.lookup_operation("login_login_access_token_post") # 选择接口
print("接口名称:", link.title)
print("接口ID:", link.name)
print("接口地址:", link.url)
print("请求方法:", link.method)
print("请求参数", link.fields)
for field in link.fields:
    pprint(to_json_schema(field.schema))
print("预期响应", link.responses)
```

 测试开发研习社

apistar 执行结果

2. 为接口构造参数

构造参数是 `hypothesis` 的看家本领,

以接口文档的请求参数的要求为策略, 即可生成随机、且符合接口要求的随机参数

hypothesis 代码示例

hypothesis 执行结果

从结果可以看到：参数名符合接口文档、参数值是随机字符的请求参数，已经构建成功

此外，还有两点需要注意：

1. 有时生成的参数，会增加额外的随机字段，但是也是符合要求的
2. 使用 `example` 的方式生成参数比较低效只适用于调试，实际执行时应该使用 `@given` 的方式来使用

```
1  @given(api_st)
2  @settings(max_examples=10)
3  def show_data(data):
4      print(data)
5      print("-" * 40)
```

更快更多的生成用例

3. 发送请求 & 断言响应

apistar 集成了请求和断言，所以这个步骤非常简单

代码示例

执行效果

此时接口的访问日志

从日志中可以看到，接口在返回非预期的结果后，代码没有马上停下来，而是继续执行，然后才报告错误

这个我后面再研究一下，看怎么样才能像测试框架一样支持 fastfail，也就是**第一个用例失败的时候就停下来**

4. 封装为 BaseCase

底层逻辑基本打通，接下来进行封装，这样在对项目进行测试时，只需用做两件事就可以自动化测试了：

1. 继承 BaseCase
2. 指定 openapi 地址

使用框架

执行效果(可点击放大)

父类代码

至此，UnitTest + PBT 的接口测试用例批量生成就封装完成了。

如果你有更好的想法，欢迎留言交流。

原创不易，喜欢请**星标+点赞+在看**，关注公众号《测试开发研习社》，不错过技术干货，谢谢鼓励！

收录于合集 #自动化测试框架 7

上一篇

【Tavern】3.tavern发送复杂HTTP请求

下一篇

让excel测试用例文件实现参数化的思路

喜欢此内容的人还喜欢

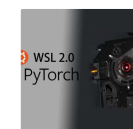
FastAPI项目：权限管理系统

7y记



MOSS的语音合成训练

谋机SEC



软件测试大杂烩（一）

芦荟全栈测开

