



链滴

pytest、tox、Jenkins 实现 python 接口自动化持续集成

作者: [zyjImmortal](#)

原文链接: <https://ld246.com/article/1584285034423>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



pytest介绍

pytest是一款强大的python测试工具，可以胜任各种级别的软件测试工作，可以自动查找测试用例并执行，并且有丰富的基础库，可以大幅度提高用户编写测试用例的效率，具备可扩展性，用户自己也可以写插件实现特定的功能，也可以安装第三方插件，非常容易的与其他工具集成到一起，比如持续集成接口自动化测试等。小编之前也写过几篇关于pytest的文章，可以到主页查看哈。

tox介绍

tox是一个命令行工具，允许测试在多种环境下执行，tox不仅能测试不同的python版本，还可以用来测试不同的依赖配置和不同的操作系统的配置。

工作原理大致是，通过setup.py文件为待测程序创建源码安装包，它会查看tox.ini中的所有环境设置并针对每个环境执行如下操作：

1. 在tox目录下创建一个虚拟环境
2. 使用pip安装依赖包
3. 使用pip在步骤1的虚拟环境中安装自己的程序包
4. 运行测试用例

下面通过一个实例来看一下tox的运行过程：

在项目根目录下新建tox.ini文件，然后加入下面的配置

```
[tox]
envlist = py36
```

```
[testenv:dev]
; deps告诉tox确保pytest已经安装，如果有多个测试依赖，可以按行罗列，同时也可以指定版本
```

```

deps =
    pytest
; 告诉tox在每个测试环境里运行pytest
commands = pytest --junitxml=junit-{envname}.xml
[pytest]
markers =
    smoke: 用于冒烟测试
get

addopts = -rsxX -l --tb=short --strict
xfail_strict = true
minversion = 3.0
norecursedirs = .* venv src *.egg dist build
testpaths = tests
python_classes = *Test Test* *Suite

```

在项目根目录下执行tox -e dev查看输出内容。

```

+ tasks_proj git:(master) * tox -e dev
GLOB sdist-make: /Users/zhouyajun/workspace/apitest/tasks_proj/setup.py
dev create: /Users/zhouyajun/workspace/apitest/tasks_proj/.tox/dev
dev installdeps: pytest
dev inst: /Users/zhouyajun/workspace/apitest/tasks_proj/.tox/.tmp/package/1/tasks-0.1.0.zip
dev installed: You are using pip version 10.0.1, however version 20.0.2 is available. You should consider upgrading via
lick=7.1.1,importlib-metadata=1.5.0,more-itertools=8.2.0,packaging=20.3,pluggy=0.13.1,py=1.8.1,pyparsing=2.4.6,py
dth=0.1.8,zip=3.1.0
dev run-test-pre: PYTHONHASHSEED='1013632314'
dev run-test: commands[0] | pytest --junitxml=junit-dev.xml

===== test session starts =====
platform darwin -- Python 3.6.6, pytest-5.4.1, py-1.8.1, pluggy-0.13.1
cachedir: .tox/dev/.pytest_cache
rootdir: /Users/zhouyajun/workspace/apitest/tasks_proj, inifile: tox.ini, testpaths: tests
collected 17 items

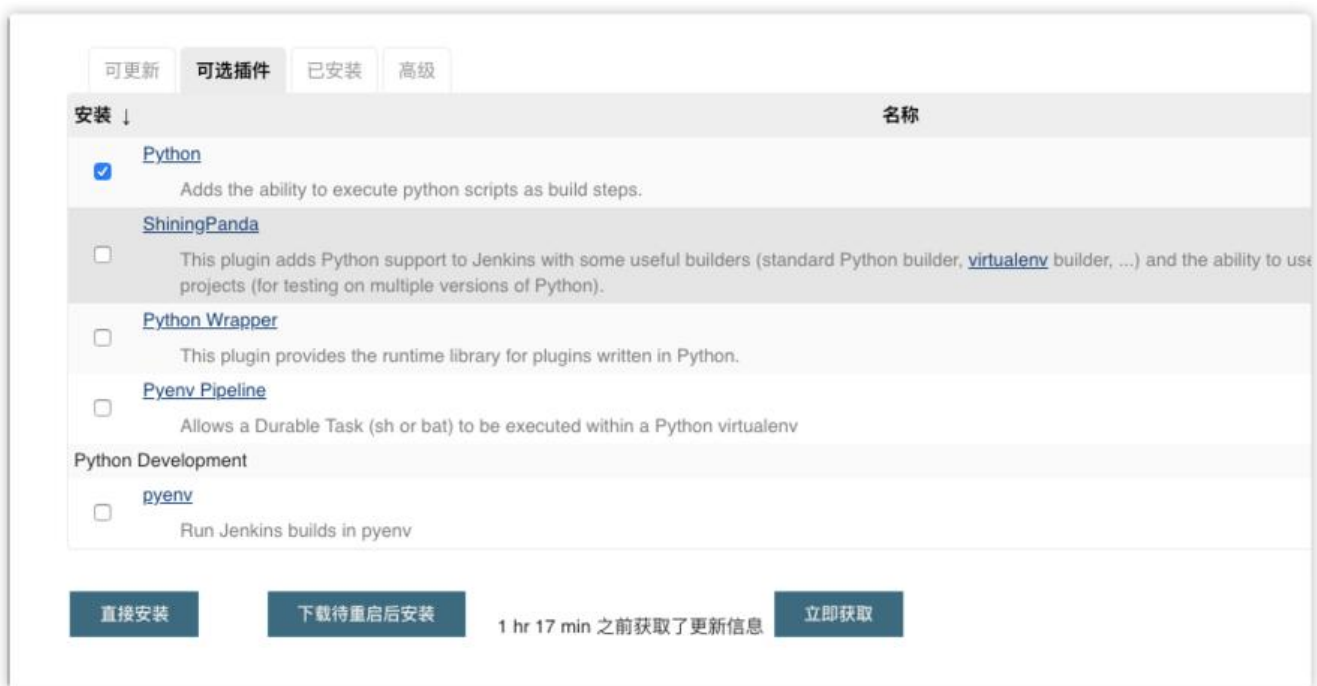
tests/func/test_add.py . [ 5%]
tests/func/test_api_exceptions.py ....F [ 35%]
tests/func/test_parallel.py .. [ 47%]
tests/func/test_unique_id.py sx [ 58%]
tests/func/test_unique_id2.py . [ 64%]
tests/func/test_unique_id3.py xxF [ 82%]
tests/unit/test_task.py ... [100%]

===== FAILURES =====
_____ TestAdd.test_done_not_bool _____
tests/func/test_api_exceptions.py:31: in test_done_not_bool
    tasks.add(Task(summary='summary', done='True'))
E   Failed: DID NOT RAISE <class 'ValueError'>
self = <func.test_api_exceptions.TestAdd object at 0x10af4c518>
_____ test_unique_id_not_a_duck _____
[XPASS(strict)]
----- Captured stdout call -----
1

```

Jenkins配置

- 安装python插件



- 创建一个多配置风格的job



- 源码配置

- 配置axes

- 添加python构建脚本

```
import toxos.chdir(os.getenv("WORKSPACE"))tox.cmdline()
```

然后构建执行，就可以了。

这篇文章说的比较粗糙，后面会加tox的介绍和Jenkins的配置说明