pytest 简介

极客时间

扫码试看/订阅

《Selenium自动化测试实战》视频课程

pytest 简介

- pytest 简介
- 安装 pytest
- pytest 官网
- 编写规则
- Console 参数介绍
- 实例
- 执行测试

pytest 简介

- pytest 是一个非常成熟的全功能的 Python 测试框架，主要特点有以下几点：
  - 简单灵活，容易上手，文档丰富；
  - 支持参数化，可以细粒度地控制要测试的测试用例；
  - 能够支持简单的单元测试和复杂的功能测试，还可以用来做 selenium/appnium 等自动化测试、接口自动化测试（pytest+requests）；
  - pytest具有很多第三方插件，并且可以自定义扩展，比较好用的如pytest-selenium（集成selenium）、pytest-html（完美html测试报告生成）、pytest-rerunfailures（失败case重复执行）、pytest-xdist（多CPU分发）等；
  - 测试用例的 skip 和 xfail 处理；
  - 可以很好的和 CI 工具结合，例如 Jenkins

pytest 安装和官网

- 安装 pytest
  - pip install pytest
- pytest 官网
  - https://docs.pytest.org/en/stable/

编写规则

- 测试文件以 test 开头（以 test 结尾也可以）
- 测试类以 Test 开头，并且不能带有 init 方法
- 测试函数以 test 开头
- 断言使用基本的 assert 即可

Console 参数介绍

- -v 用于显示每个测试函数的执行结果
- -q 只显示整体测试结果
- -s 用于显示测试函数中 print() 函数输出
- -x, --exitfirst，在第一个错误或测试失败时立即退出
- -h 帮助

执行测试

- 配置PyCharm执行：
  - Tools->Python Integrated tools-> Default test runner
- main 方法
  - pytest.main(['-s', '-v', '01-pytest简介.py'])
- 命令行
  - pytest -s -v test.py

pytest 标记

Pytest 标记

- Pytest 查找测试策略
- 标记测试函数

Pytest 查找测试策略

- 默认情况下，pytest 会递归查找当前目录下所有以 test 开始或结尾的 Python 脚本，
- 并执行文件内的所有以 test 开始或结束的函数和方法。
- 实例

标记测试函数

- 由于某种原因（如 test_func2 的功能尚未开发完成），我们只想执行指定的测试函数。在 pytest 中有几种方式可以解决：
- 第一种，显式指定函数名，通过 :: 标记
  - test_no_mark.py::test_func1
- 第二种，使用模糊匹配，使用 -k 选项标识。
  - pytest -k func1 test_no_mark.py
- 第三种，使用 pytest.mark 在函数上进行标记

给用例打标签

- 注册标签名，通过 .ini 配置文件，格式如下：
  - [pytest]
  - markers =
  - do: do
  - undo: undo
- 在用例上打标记
- 实例演示

pytest 参数化处理

pytest 参数化处理

- 在 pytest 中，也可以使用参数化测试，即每组参数都独立执行一次测试。
- 使用的工具就是 pytest.mark.parametrize(argnames, argvalues)。

数据格式

- 数据格式可以是：
  - 列表
  - 元组
  - 字典

增加可读性

- 参数化装饰器有一个额外的参数 ids，可以标识每一个测试用例，自定义测试数据结果的显示，为了增加可读性，
- 我们可以标记每一个测试用例使用的测试数据是什么，适当的增加一些说明
- 实例演示

自定义id做标识

- 除了使用 ids 参数增加输出可读性外，我们还可以在参数列表的参数旁边定义一个 id 值来做标识

- 看下面实例

pytest fixture

Pytest fixture

- 定义 fixture 跟定义普通函数差不多，唯一区别就是在函数上加个装饰器 @pytest.fixture()
- fixture 命名不要以 test 开头，跟用例区分开。fixture 是有返回值得，没有返回值默认为 None。
- 用例调用 fixture 的返回值，直接就是把 fixture 的函数名称当做变量名称。
- 实例演示

Pytest setup 和 teardown

unittest setup 和 teardown 简介

- 学过 unittest 的都知道里面用前置和后置setup和teardown 非常好用。
- 在每次用例开始前和结束后都去执行一次。
- 当然还有更高级一点的 setupClass 和 teardownClass，需配合 @classmethod 装饰器一起使用。
- 在做 selenium 自动化的时候，它的效率尤为突出，可以只启动一次浏览器执行多个用例。

pytest setup 和 teardown 简介

- 模块级（setup_module/teardown_module）开始于模块始末，全局的
- 函数级（setup_function/teardown_function）只对函数用例生效（不在类中）
- 类级（setup_class/teardown_class）只在类中前后运行一次(在类中)
- 方法级（setup_method/teardown_method）开始于方法始末（在类中）
- 类里面的（setup/teardown）运行在调用方法的前后

pytest allure 生成测试报告

pytest allure 生成测试报告

- 安装
  - pip install allure-pytest
- 官方文档
  - https://docs.qameta.io/
- 下载 allure
  - https://dl.bintray.com/qameta/generic/io/qameta/allure/allure/2.7.0/

allure 用例描述

使用方法	参数值	参数说明
@allure.epic()	epic描述	敏捷里面的概念，定义史诗，往下是feature
@allure.feature()	模块名称	功能点的描述，往下是story
@allure.story()	用户故事	用户故事，往下是title
@allure.title(用例的标题)	用例的标题	重命名html报告名称
@allure.testcase()	测试用例的链接地址	对应功能测试用例系统里面的case
@allure.issue()	缺陷	对应缺陷管理系统里面的链接
@allure.description()	用例描述	测试用例的描述
@allure.step()	操作步骤	测试用例的步骤
@allure.severity()	用例等级	blocker，critical，normal，minor，trivial
@allure.link()	链接	定义一个链接，在测试报告展现
@allure.attachment()	附件	报告添加附件

使用 pytest 重构项目

使用 pytest 重构项目

- 继承 unittest.TestCase 修改为继承 object
- unittest setup 方法修改为 pytest setup
- unitest 的断言修改为 Python 断言 assert
- 使用 pytest 依赖插件
- pip3 install pytest-dependency

极客时间

扫码试看/订阅

《Selenium自动化测试实战》视频课程