

# unittest 简介



扫码试看/订阅

《Selenium自动化测试实战》视频课程

# 什么是 unittest

- unittest 是 Python 单元测试框架，类似于 JUnit 框架。
- unittest 中有5个重要的概念：test fixture, test case, test suite, test runner, test loader

# unittest 重要概念

- Testcase
  - 一个 TestCase 的实例就是一个测试用例。什么是测试用例呢？就是一个完整的测试流程，包括测试前准备环境的搭建（setUp），执行测试代码（run），以及测试后环境的还原（tearDown）。单元测试（unit test）的本质也就在这里，一个测试用例是一个完整的测试单元，通过运行这个测试单元，可以对某一个问题进行验证。
- Test suite
  - 多个测试用例集合在一起，就是 TestSuite，而且 TestSuite 也可以嵌套 TestSuite。

# unittest 重要概念

- Test runner
  - 是来执行测试用例的，其中的 `run(test)` 会执行 `TestSuite/TestCase` 中的 `run(result)` 方法。
- TestLoader
  - 是用来加载 `TestCase` 到 `TestSuite` 中的，其中有几个 `loadTestsFrom__()` 方法，就是从各个地方寻找 `TestCase`，创建它们的实例，然后 `add` 到 `TestSuite` 中，再返回一个 `TestSuite` 实例。
- Test fixture
  - 对一个测试用例环境的搭建和销毁，是一个 `fixture`，通过覆盖 `TestCase` 的 `setUp()` 和 `tearDown()` 方法来实现。

# unittest 实例演示

- 实例演示

# unittest 中的方法

# setup 和 teardown

- 方法分类
  - 分为类方法,类方法只执行一次
  - 实例方法，实例方法每次运行用例都执行。
- 执行顺序
  - setUpClass->setUp->testA->tearDown->setUp->testB>tearDown->tearDownClass
  - 用例之间按用例名称ASCII码的顺序加载，数字与字母顺序为0~9，A~Z，a~z。



# 断言方法

方法	检查
<code>assertEqual(a, b)</code>	<code>a == b</code>
<code>assertNotEqual(a, b)</code>	<code>a != b</code>
<code>assertTrue(x)</code>	<code>bool(x) is True</code>
<code>assertFalse(x)</code>	<code>Bool(x) is False</code>
<code>assertIs(a, b)</code>	<code>a is b</code>
<code>assertIsNot(a, b)</code>	<code>a is not b</code>
<code>assertIsNone(x)</code>	<code>x is None</code>
<code>assertIsNotNone(x)</code>	<code>x is not None</code>
<code>assertIn(a, b)</code>	<code>a in b</code>
<code>assertNotIn(a, b)</code>	<code>a not in b</code>
<code>assertIsInstance(a, b)</code>	<code>isinstance(a,b)</code>
<code>assertNotIsInstance(a, b)</code>	<code>not isinstance(a,b)</code>

# 测试用例、测试套件和测试加载器

# 加载测试用例的四种方法

- 实例化
  - `suite = unittest.TestSuite()`
  - `loader = unittest.TestLoader()`
- 方法一： 通过测试用例类进行加载
  - `suite.addTest(loader.loadTestsFromTestCase(MyTest1))`
  - `suite.addTest(loader.loadTestsFromTestCase(MyTest2))`

# 加载测试用例的四种方法

- 方法二： 通过测试用例模板去加载
  - `suite.addTest(loader.loadTestsFromModule(MyTest1))`
  - `suite.addTest(loader.loadTestsFromModule(MyTest2))`
- 方法三： 通过路径加载
  - `import os`
  - `path = os.path.dirname(os.path.abspath(__file__))`
  - `suite.addTest(loader.discover(path))`

# 加载测试用例的四种方法

- 方法四：逐条加载测试用例 low
  - `case1 = MyTest1("test1")`
  - `case2 = MyTest1("test2")`
  - `suite.addTest(case1)`
  - `suite.addTest(case2)`

# 使用 unittest 重构测试用例

# 重构步骤

- 拷贝上一节测试用例
- 习惯测试用例基础 `unittest.TestCase`
- 将 `__init__` 修改为 `setup`
- 将 `assert` 断言, 修改为 `unittest`断言
- 关掉浏览器放到 `teardown`
- 注意: 第二次测试情况数据
- 注意: 顺序测试要加序号, 例如: `test1`、`test2`

# 实例演示

- 实例演示



为项目添加测试报告



扫码试看/订阅

《Selenium自动化测试实战》视频课程