

Page Object 飞书实战

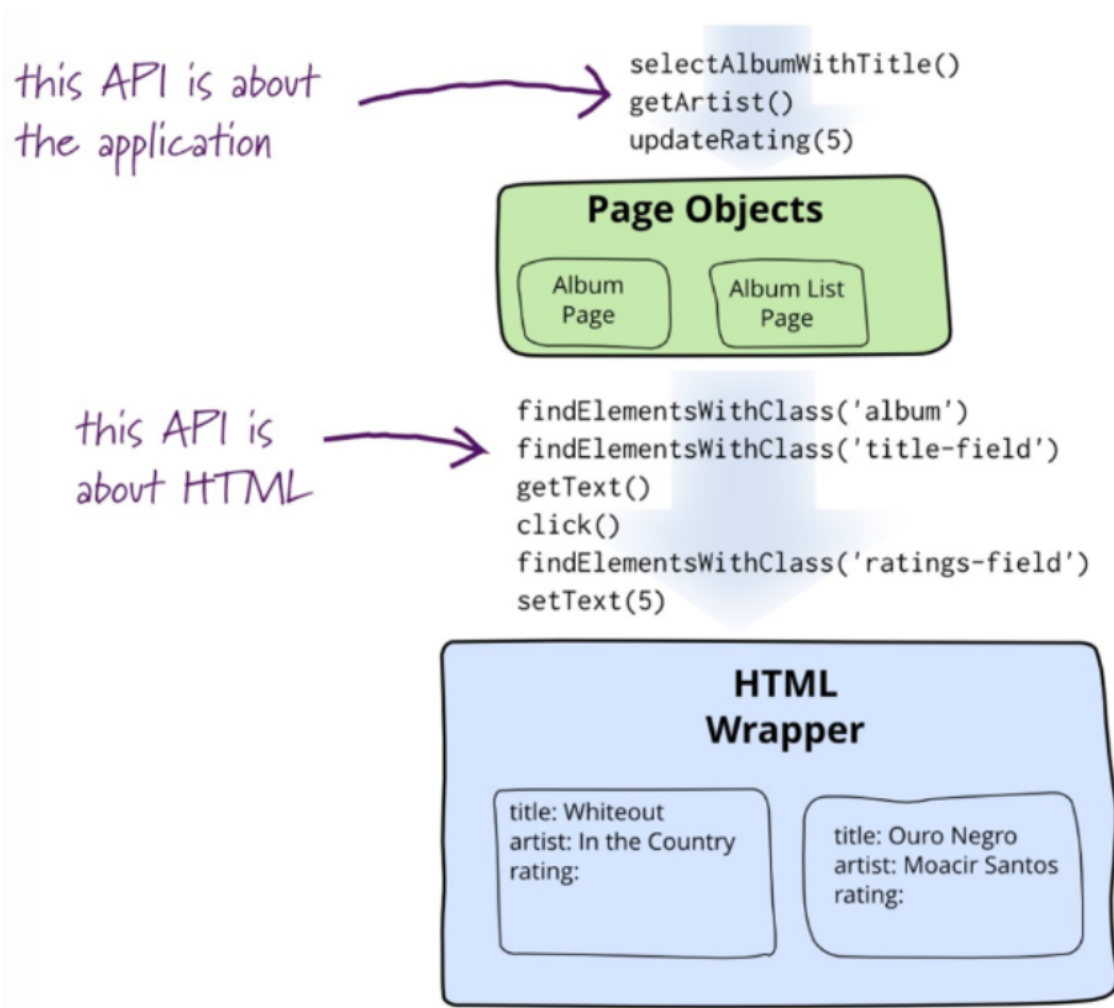
1 Page Object 简介

1.1 概念

PO是 Page Object 的缩写，PO模式是 UI 自动化测试项目开发实践的最佳设计模式之一

核心思想是通过对界面元素的封装减少冗余代码，同时在后期维护中，若元素定位发生变化，只需要调整页面元素封装的代码，提高测试用例的可维护性、可读性

1.2 认识po



做法

页面为单位独立建模

隐藏实现细节

本质是面向接口编程

即使改变具体的元素，**page**对象的接口不应当发生变化

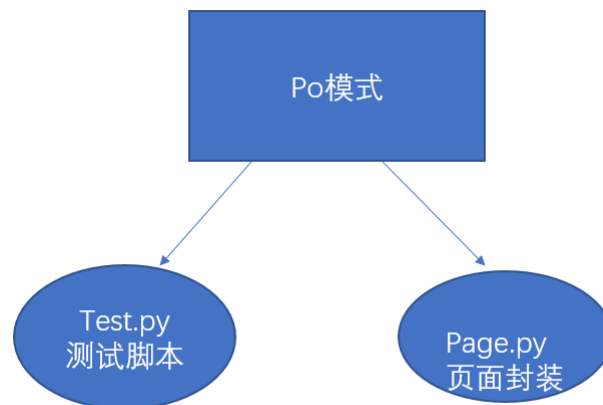
例：如果你要操作输入框，那么只需调用 `search_input()` 方法并传入搜索关键字即可，并不需要关心输入框是如何定位的

原则

用公共方法代表 **UI** 所提供的功能

点击按钮后有新的页面跳转，使用**return**方法表示跳转

只对页面重要的元素进行**po**设计



2 实现Page Object

项目准备：<https://www.feishu.cn/>

需求：1、首页点击登录-点击立即注册

2、首页点击免费使用，输入手机号码'18888888888'

样板代码实现

代码示例

```
from selenium import webdriver
import time

class TestDemo:
    # 初始化信息
    def setup(self):
        self.driver = webdriver.Chrome()
        self.driver.get('https://www.feishu.cn/')
        self.driver.implicitly_wait(5)

    # 点击登录 -- 切换窗口 -- 输入手机号
```

```

def test_case_login(self):
    self.driver.find_element_by_link_text('登录').click()
    self.driver.switch_to.window(self.driver.window_handles[-1])

    self.driver.find_element_by_xpath("//div[@class='login-go-
registe']/span/span").click()
    time.sleep(3)
    self.driver.switch_to.window(self.driver.window_handles[-1])

self.driver.find_element_by_id('register_phone').send_keys('18888888888')

# 点击免费使用 -- 切换窗口 -- 输入手机号
def test_case_free(self):
    self.driver.find_element_by_link_text('免费使用').click()
    self.driver.switch_to.window(self.driver.window_handles[-1])

self.driver.find_element_by_id('register_phone').send_keys('18888888888')

def teardown(self):
    self.driver.quit()

```

存在大量重复性代码，需要po设计解决

2.1 抽取公共基类

page新建base文件 存放页面文件的公共类

创建base.py文件

```

from selenium import webdriver

import time

from selenium.webdriver.remote.webdriver import WebDriver

'''basepage是所有pageobject的父类， 为子类提供公共方法， 比如初始化 driver 和退出driver
代码在base_page模块的basepage类使用__init__初始方法进行初始化操作，包括driver的复用
driver的赋值、全局的等待时间设置 == 常见隐式等待'''

# 创建base基类
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.remote.webdriver import WebDriver
import time
# webdriver 定义 标识说明

from selenium.webdriver.support.wait import WebDriverWait

class Base:

```

```

def __init__(self,driver :WebDriver=None):
    # 传入数据 driver 以及driver 初始化 做对应的判断
    if driver is None: # 此处对driver进行复用, 如果不存在driver, 就构造一个新的
        self.driver = webdriver.Chrome()
        self.driver.implicitly_wait(5)
        self.driver.get('https://www.feishu.cn/')

    else:# 如果driver有值, 比如login 和 free 页面需要这个方法, 避免重复构造driver
        self.driver = driver

def close(self):
    time.sleep(5)
    self.driver.quit()

```

2.2 抽取page文件

- 按照po原则, 设计三个页面首页、登录、免费使用, 创建对应文件
- 区分操作细节和操作步骤 - 项目新建page包和testcase包
- page新建首页、登录、免费使用三个页面文件
- testcase新建testcase文件

创建index.py

```

# 进入首页 根据功能 新建两个方法  goto_login  goto_register

# 导入base
import time

from selenium.webdriver.common.by import By

from selenium3.page.base import Base
from selenium3.page.register import Register
from selenium3.page.login import Login

class Index(Base):

    # 点击登陆
    def goto_login(self):
        self.driver.find_element_by_link_text('登录').click()
        # 返回实例化 login
        self.driver.switch_to.window(self.driver.window_handles[-1])
        return Login(self.driver)

    # 点击免费使用
    def goto_register(self):
        self.driver.find_element_by_link_text('免费使用').click()

        # 返回实例化的register
        self.driver.switch_to.window(self.driver.window_handles[-1])
        return Register(self.driver)

```

创建login.py

```
# 登陆后 页面展示 返回为注册页
```

```
from selenium3.page.base import Base
from selenium3.page.register import Register
import time

class Login(Base):

    def registry(self):
        self.driver.find_element_by_xpath("//div[@class='login-go-
register']/span/span").click()

        time.sleep(3)
        # 点击立即注册 跳转到免费使用页面

        return Register(self.driver)
```

创建register.py

```
# 注册页
from selenium.webdriver.common.by import By

from selenium3.page.base import Base

class Register(Base):

    def register(self,phone):

self.driver.find_element_by_id('register_phone').send_keys('18888888888')

        return self
```

2.3 抽取find、click、input、get_text、handles方法

base文件创建对应方法

```
# 创建base基类
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.remote.webdriver import WebDriver
import time
# webdriver 定义 标识说明
# 构建
from selenium.webdriver.support.wait import WebDriverWait

class Base:

    def __init__(self,driver :WebDriver=None):
```

```

# 传入数据 driver 以及driver 初始化 做对应的判断
if driver is None:
    self.driver = webdriver.Chrome()
    self.driver.implicitly_wait(5)
    self.driver.get('https://www.feishu.cn/')

else:
    self.driver = driver

def close(self):
    time.sleep(5)
    self.driver.quit()

# 提取find方法

def find(self,by,loc):

    return WebDriverWait(self.driver,timeout=5).until(lambda x
:self.driver.find_element(by,loc))

# 提取click方法

def click(self,by,loc):
    return self.find(by,loc).click()

# 提取输入方法

def keys(self,by,loc,v):
    return self.find(by,loc).send_keys(v)

# 提取文本信息

def get_text(self,by,loc):
    return self.find(by,loc).text

# handles方法

def handles(self>window):
    return self.driver.switch_to.window(window)

```

index.py 提取base中 find、click、input、handles方法使用

```

# 进入首页 根据功能 新建两个方法    goto_login    goto_register

# 导入base
import time

from selenium.webdriver.common.by import By

from selenium3.page.base import Base
from selenium3.page.register import Register
from selenium3.page.login import Login

```

```

class Index(Base):

    # 点击登录
    def goto_login(self):
        # self.driver.find_element_by_link_text('登录').click()
        # 抽取find方法
        # self.find(By.LINK_TEXT, '登录').click()
        # 抽取click方法
        self.click(By.LINK_TEXT, '登录')

        # 返回实例化 login
        # self.driver.switch_to.window(self.driver.window_handles[-1])
        # 使用frame方法
        self.handles(self.driver.window_handles[-1])
        return Login(self.driver)

    # 点击免费使用

    def goto_register(self):
        # self.driver.find_element_by_link_text('免费使用').click()
        # click方法调用

        self.click(By.LINK_TEXT, '免费使用').click()

        time.sleep(3)

        # 返回实例化的register
        # self.driver.switch_to.window(self.driver.window_handles[-1])
        self.handles(self.driver.window_handles[-1])
        return Register(self.driver)

```

login.py 和 register.py两个页面调用方法同index一致

2.4 参数化

项目目录新建data 文件格式yaml

执行文件testcase.py 采用 @pytest.mark.parametrize()方法

```

# index 页面进行组装操作    不传入细节信息

from selenium3.page.index import Index
import time
import pytest
import yaml

class Test_Index:

    # 初始化

    def setup(self):

```

```

        self.index = Index()

    # 登录功能
    @pytest.mark.parametrize('phone',
yaml.safe_load(open('../data/data/data.yml')))
    def test_login(self, phone):
        self.index.goto_login().registry().register(phone)

    # 免费使用
    @pytest.mark.parametrize('phone',
yaml.safe_load(open('../data/data/data.yml')))
    def test_register(self, phone):
        self.index.goto_register().register(phone)

def teardown(self):
    time.sleep(5)
    self.index.close()

```

2.5 生成测试报告

2.5.1 pytest -- html报告

r.html

Report generated on 19-Mar-2021 at 15:01:13 by [pytest-html](#) v2.1.1

Environment

JAVA_HOME	D:\developer tools\Java\jdk1.8.0_181
Packages	{"pluggy": "0.13.1", "py": "1.8.1", "pytest": "5.4.3"}
Platform	Windows-10-10.0.17134-SP0
Plugins	{"allure-pytest": "2.8.16", "html": "2.1.1", "metadata": "1.8.0", "ordering": "0.6", "rerunfailures": "9.0"}
Python	3.7.2

Summary

2 tests ran in 28.42 seconds.

(Un)check the boxes to filter the results.

☒ 2 passed, ☒ 0 skipped, ☒ 0 failed, ☒ 0 errors, ☒ 0 expected failures, ☒ 0 unexpected passes, ☒ 0 rerun

Results

[Show all details](#) / [Hide all details](#)

Result	Test
Passed (show details)	test_case.py::Test_case::test_login[123]
Passed (show details)	test_case.py::Test_case::test_login[567]

2.5.2 Allure报告

执行命令

```

pytest --alluredir=report test_case.py

allure generate report/ -o report/html --clean

```




Allure

功能

总览

类别

测试套

图表

时间刻度

功能

包

通过用例状态过滤: 00200 Marks: [icon]

order	名称	用时	状态
✓ #1	test_login	13s 025ms	
✓ #2	test_free	1s 501ms	