

# Page Object 设计模式



扫码试看/订阅

《移动端自动化测试实战》视频课程

# Page Object Model (POM) 简史

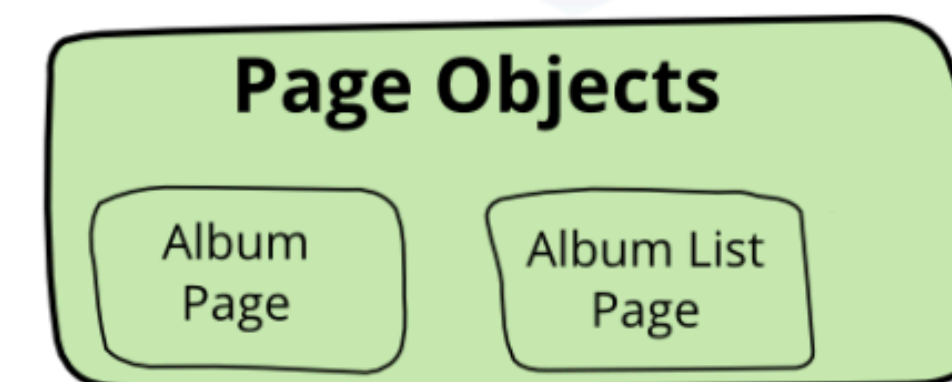
- 2013 Martin Fowler <https://martinfowler.com/bliki/PageObject.html>
- 2015 Selenium <https://github.com/SeleniumHQ/selenium/wiki/PageObjects>
- Selenium Python Client <https://selenium-python.readthedocs.io/page-objects.html>
- Mozilla PyPOM <https://pypom.readthedocs.io/en/latest/>

# Page Object 模式

- 做法
  - 以页面为单位独立建模
  - 隐藏实现细节
  - 本质是面向接口编程
- 优点
  - 减少重复 find click 样板代码
  - 易读性提高
  - 页面修改不影响测试用例
- Image From Martin Fowler

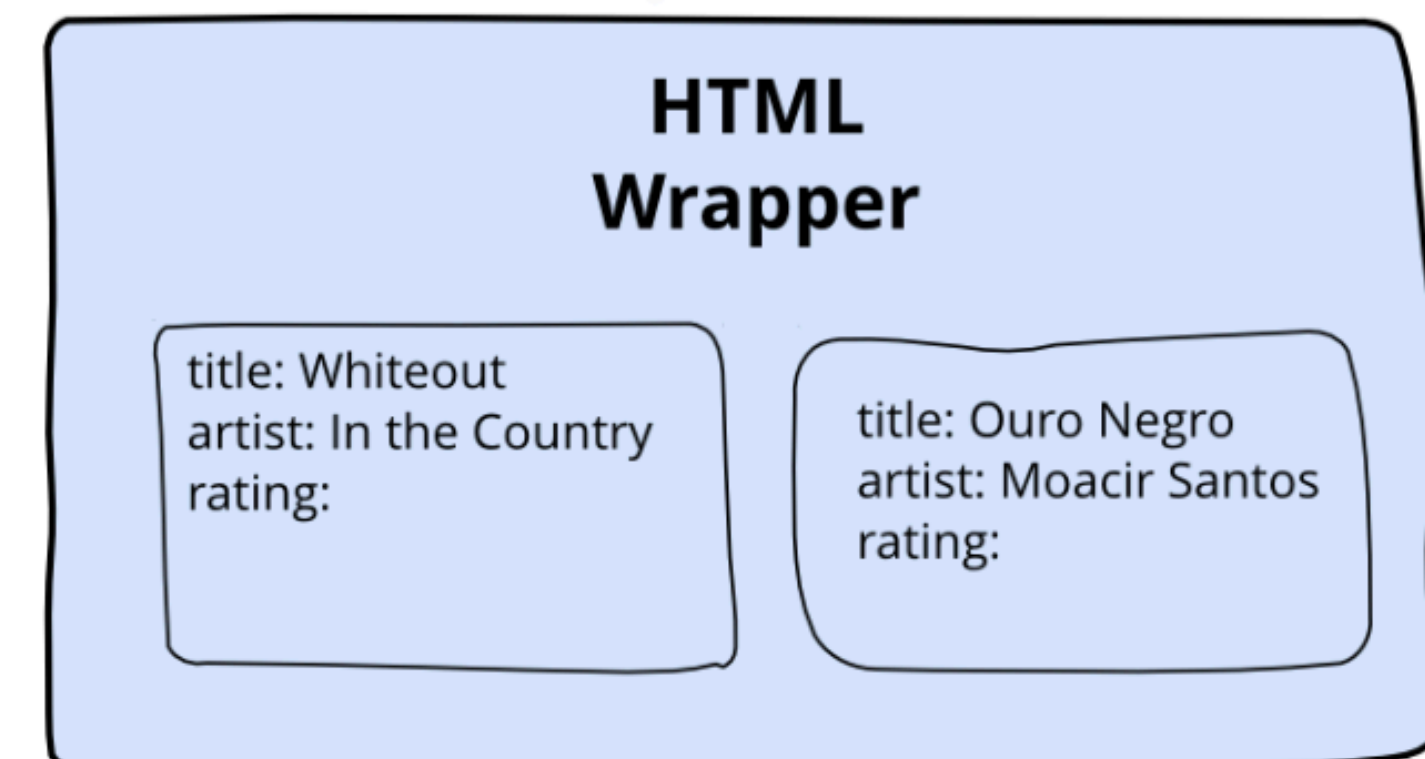
this API is about  
the application

selectAlbumWithTitle()  
getArtist()  
updateRating(5)



this API is  
about HTML

findElementsWithClass('album')  
findElementsWithClass('title-field')  
getText()  
click()  
findElementsWithClass('ratings-field')  
setText(5)



# Page Object 六大设计原则

# Page Object Model 的基本原则

## Summary

---

- The public methods represent the services that the page offers
- Try not to expose the internals of the page
- Generally don't make assertions
- Methods return other PageObjects
- Need not represent an entire page
- Different results for the same action are modelled as different methods

# Page Object 模式原则解读

- 方法意义
  - 用公共方法代表 UI 所提供的功能
  - 方法应该返回其他的 Page Object 或者返回用于断言的数据
  - 同样的行为不同的结果可以建模为不同的方法
  - 不要在方法内加断言
- 字段意义
  - 不要暴露页面内部的元素给外部
  - 不需要建模 UI 内的所有元素

# 登录场景

- 登陆页面提供 login findPassword 功能
  - Login 类 + login findPassword 方法
- 登录页面内的元素有多少并不关心，隐藏内部界面控件
- 登录成功和失败会分别返回不同的页面
  - findPassword
  - loginSuccess
  - loginFail
- 通过方法返回值判断登录是否符合预期



# Page Object 第一个例子

# 编写用例顺序

- 根据界面封装 page 类与方法，实现可以为空
- 编写用例，不断重构明确 page 里方法的入参和返回值
- 开始实现 page 内的方法
- 调试
- 整体类似 TDD 风格

# BasePage 封装改进

# BasePage 封装

- 实现通用的 Page 方法，对常用自动化行为做封装
- 管理各种 Driver
- 减少每个 Page 对 Appium、Selenium 等库的太多依赖

# 自动化行为封装

- 异常弹框处理：广告、好评、升级、tips 等弹框
- 通用自动化能力封装

# Page Object 模式实战

# Java Python的封装方式

- Java:
  - Page Factory + @FindBy
- Python:
  - PyDOM、独立封装

# 基于 POM 的用例组织结构

- page: 完成对页面的封装
- driver: 完成对 Web、Android、iOS、接口的驱动
- testcase: 调用各类 page 完成业务流程并进行断言
- data: 配置文件和数据驱动
- utils: 其他便捷的功能封装, 可选



# 测试用例管理

- 用例组织结构：
  - 使用 package 管理业务模块
  - 使用 class 管理业务对象、使用method完成业务具体行为
- 数据驱动：测试数据、测试步骤、测试断言
- 测试用例：
  - 使用 testcase 完成测试步骤的定义
  - 使用 assertion 完成业务正确性校验
- 持续集成：使用 Jenkins 完成持续集成



扫码试看/订阅

《移动端自动化测试实战》视频课程