

# 使用 Selenium 测试凤凰网

## 作业目标

通过本次实验,学习使用 Selenium WebDriver 自动化测试凤凰网(www.ifeng.com)的基本功能,包括页面元素定位、交互操作和断言验证。

## 作业要求

1. 使用 Python (或 Java) 和 Selenium 编写测试脚本
2. 测试至少包含凤凰网的 3 个主要功能模块
3. 包含异常处理和断言验证
4. 提交完整的代码和测试报告

## 实验步骤

### 1. 环境准备

Python

Pycharm

安装必要库

```
pip install selenium pytest
```

下载对应浏览器的 WebDriver (如 ChromeDriver) 并配置到系统 PATH 中。

### 2. 基础测试代码示例

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time

class TestPhoenixNews:
    def setup_method(self):
        self.driver = webdriver.Chrome()
        self.driver.maximize_window()
        self.driver.get("https://www.ifeng.com/")
        self.wait = WebDriverWait(self.driver, 10)

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

    def test_homepage_load(self):
        """测试首页加载"""
        assert "凤凰网" in self.driver.title
        logo = self.wait.until(EC.presence_of_element_located((By.CSS_SELECTOR, ".logo")))
        assert logo.is_displayed()

    def test_news_navigation(self):
        """测试新闻导航"""
        news_link = self.wait.until(EC.element_to_be_clickable((By.LINK_TEXT, "新闻")))
        news_link.click()
        self.wait.until(EC.title_contains("新闻"))
        assert "新闻" in self.driver.title

    def test_search_function(self):
        """测试搜索功能"""
        search_box = self.wait.until(EC.presence_of_element_located((By.CSS_SELECTOR, ".searchInput")))
        search_box.send_keys("科技")
        search_box.send_keys(Keys.RETURN)
        time.sleep(2) # 等待结果加载
        assert "科技" in self.driver.title or "搜索结果" in self.driver.title
```

### 3. 扩展任务

- (1) 测试视频模块：导航到视频页面，播放一个视频并验证播放器状态
- (2) 测试登录功能：尝试使用测试账号登录（注意：不要提交真实凭证）
- (3) 测试评论区：找到一篇文章，展开评论区并验证元素存在
- (4) 响应式测试：调整浏览器窗口大小，验证页面布局变化

### 4. 测试报告要求

- (1) 记录每个测试用例的执行结果（通过/失败）
- (2) 对失败的测试用例进行分析
- (3) 附上测试过程中捕获的截图（特别是失败情况）
- (4) 提出改进建议

### 提交内容

1. 完整的 Python 测试脚本文件
2. 测试报告文档（PDF 或 Markdown 格式）
3. 测试回放录屏

### 评分标准

1. 代码完整性（30%）：是否完成所有要求的测试场景
2. 代码质量（20%）：代码结构、可读性、异常处理
3. 测试报告（30%）：详细程度、问题分析、改进建议
4. 创新性（20%）：是否包含超出基本要求的测试场景

### 注意事项

1. 尊重网站的使用条款，不要进行高频请求
2. 测试时注意时间间隔，避免给服务器造成负担
3. 不要测试需要付费或敏感信息的功能
4. 如遇到验证码等反爬机制，应在报告中说明并跳过相关测试