Appium 架构介绍

极客时间

扫码试看/订阅

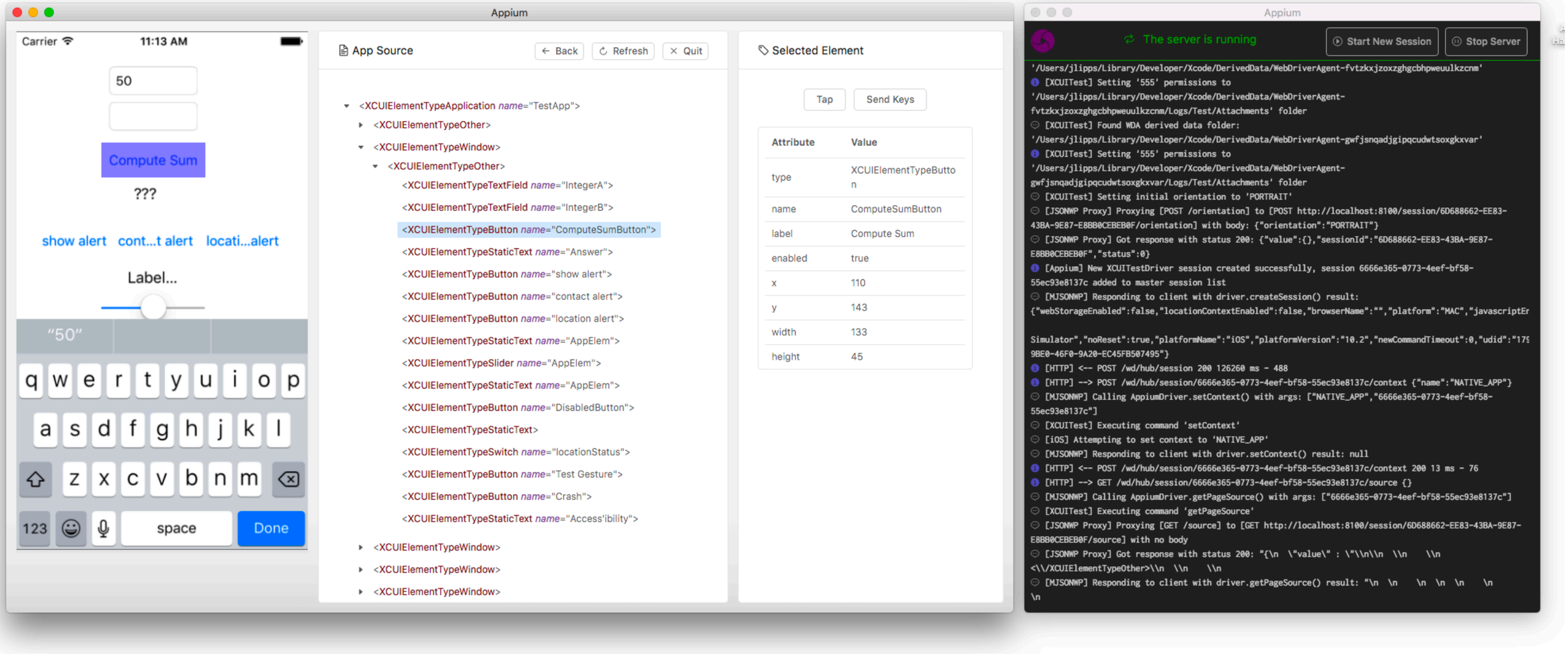
《移动端自动化测试实战》视频课程

Appium 生态工具

- adb：Android 的控制工具，用于获取 Android 的各种数据和控制
- Appium Desktop：内嵌了 Appium Server 和 Inspector 的综合工具
- Appium Server：Appium 的核心工具，命令行工具
- Appium Clients：各种语言的客户端封装库，用于连接 appium server
  - Java、Python、Ruby、robotframework-appium
- AppCrawler 自动遍历工具

Appium Desktop 安装

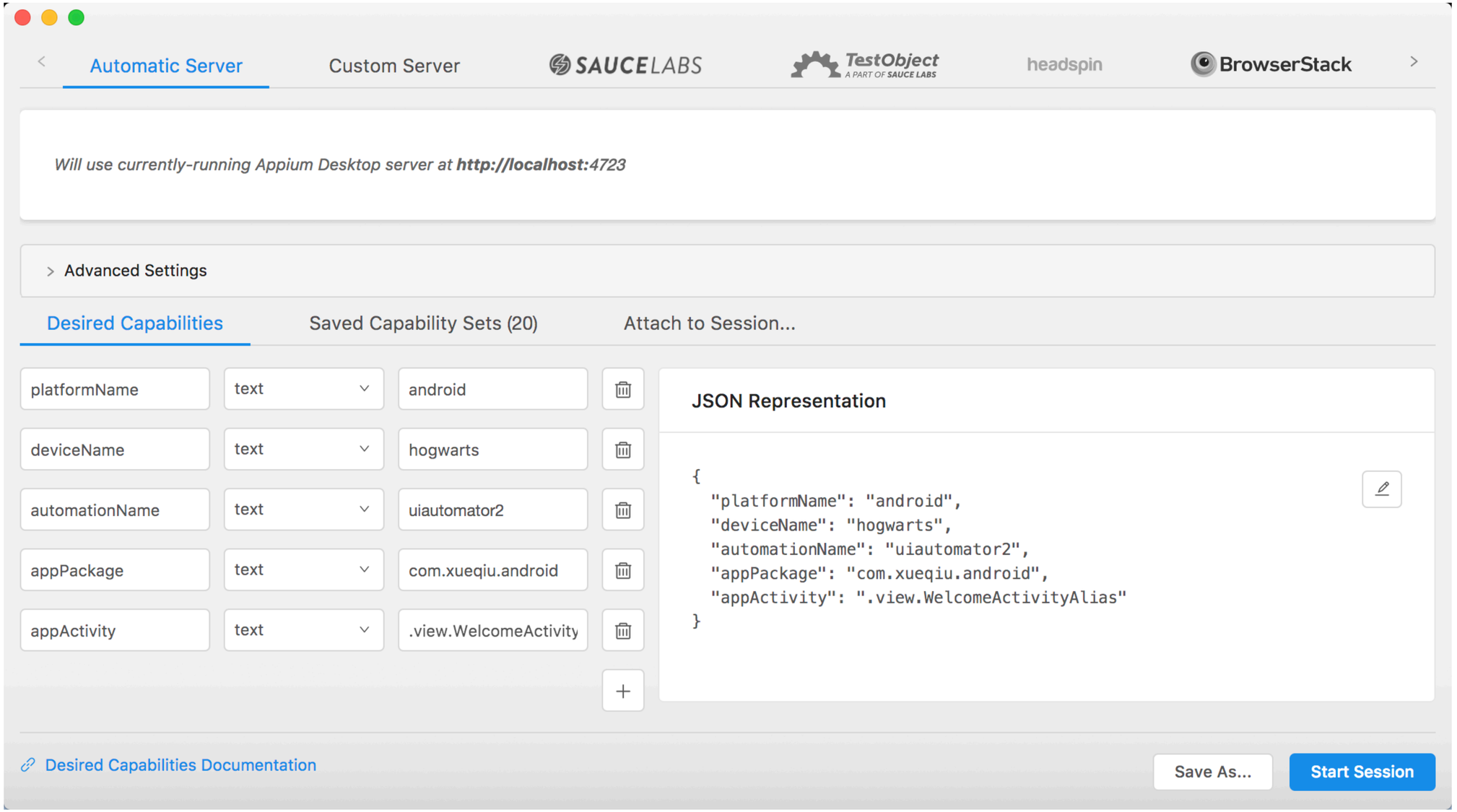
Appium Desktop 工具包



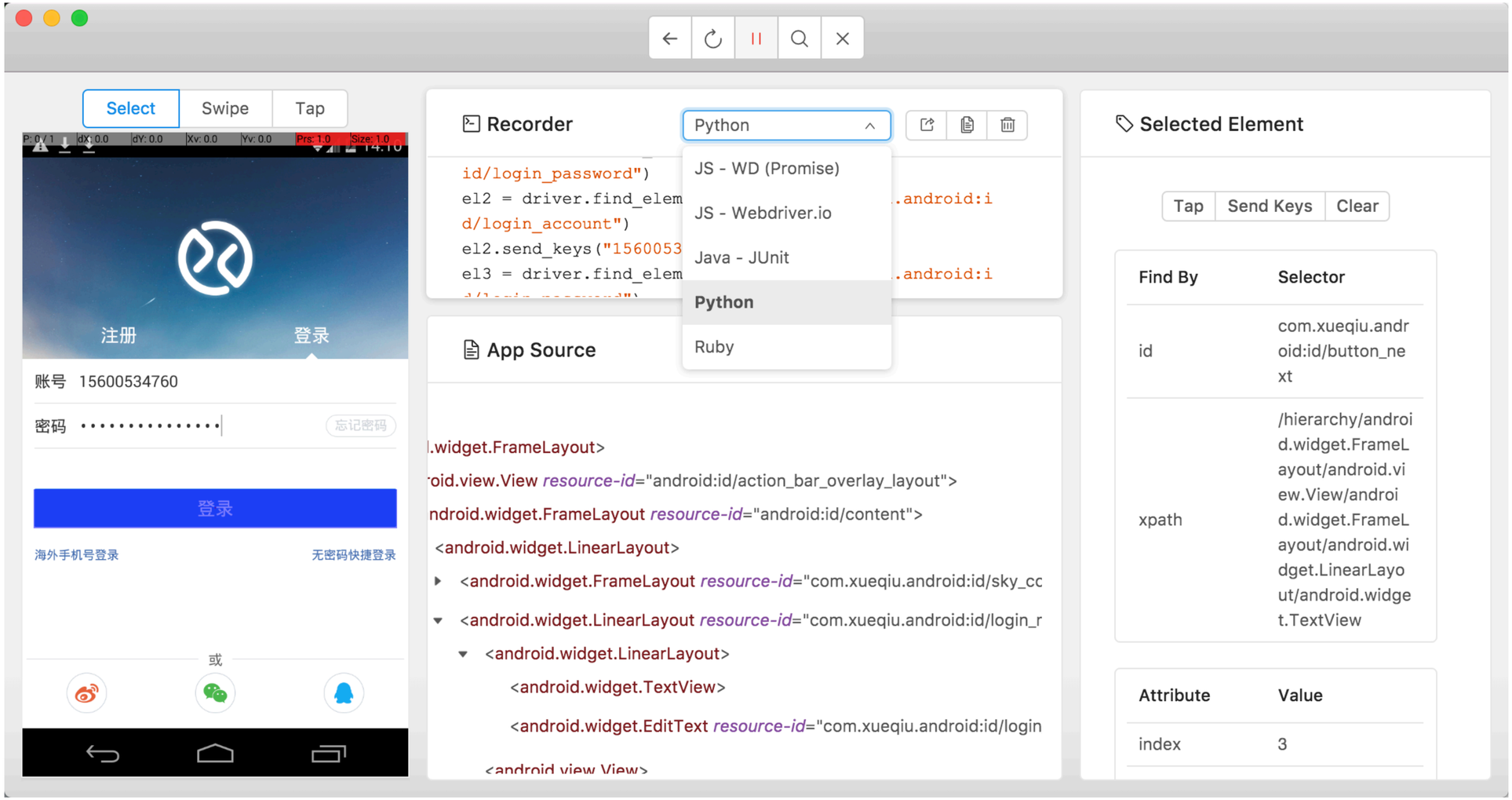
Desktop 主要功能

- UI 分析
- 录制用例
- 元素查找测试
- Attach 已有的 session
- 云测试

配置待测应用


Automatic Server
Custom Server
SAUCELABS
TestObject
A PART OF SAUCE LABS
headspin
BrowserStack
Will use currently-running Appium Desktop server at http://localhost:4723
Advanced Settings
Desired Capabilities
Saved Capability Sets (20)
Attach to Session...
platformName
text
android
deviceName
text
hogwarts
automationName
text
uiautomator2
appPackage
text
com.xueqiu.android
appActivity
text
.view.WelcomeActivity
JSON Representation
{
"platformName": "android",
"deviceName": "hogwarts",
"automationName": "uiautomator2",
"appPackage": "com.xueqiu.android",
"appActivity": ".view.WelcomeActivityAlias"
}
Desired Capabilities Documentation
Save As...
Start Session

利用 Appium Desktop 生成用例模板



Appium Server 安装

Appium Server 环境安装

- 安装 Node.js，推荐 LTS 版本
- 安装 Appium
  - 官方文档的安装方式（因为服务器在海外，因为众所周知的原因，基本安装不上）：
    - npm install -g appium
  - 淘宝 cnpm （最稳定的方法）
    - npm install -g cnpm --registry=https://registry.npm.taobao.org
    - cnpm install -g appium
- 相关链接：https://shimo.im/docs/HCt3J9DgHwCQ8HdD/

为什么 Appium Server 安装困难

- 部分依赖在海外无法访问。
- Node.js 安装不要用 root 权限安装。
- Node.js 版本不宜太低，也不宜太高，推荐 LTS 版本，或者从 Appium 源代码中获得推荐的版本。
- 需要 python2 存在。
- node_module 不具备写入权限。
- Windows 下需要依赖编译工具。
- PATH 变量要设置 adb java 等路径，可以用 appium-doctor 验证。
- iOS 还需要解决 WebDriverAgent 的编译和依赖下载问题。

Appium Desktop 用例录制

录制用例并执行

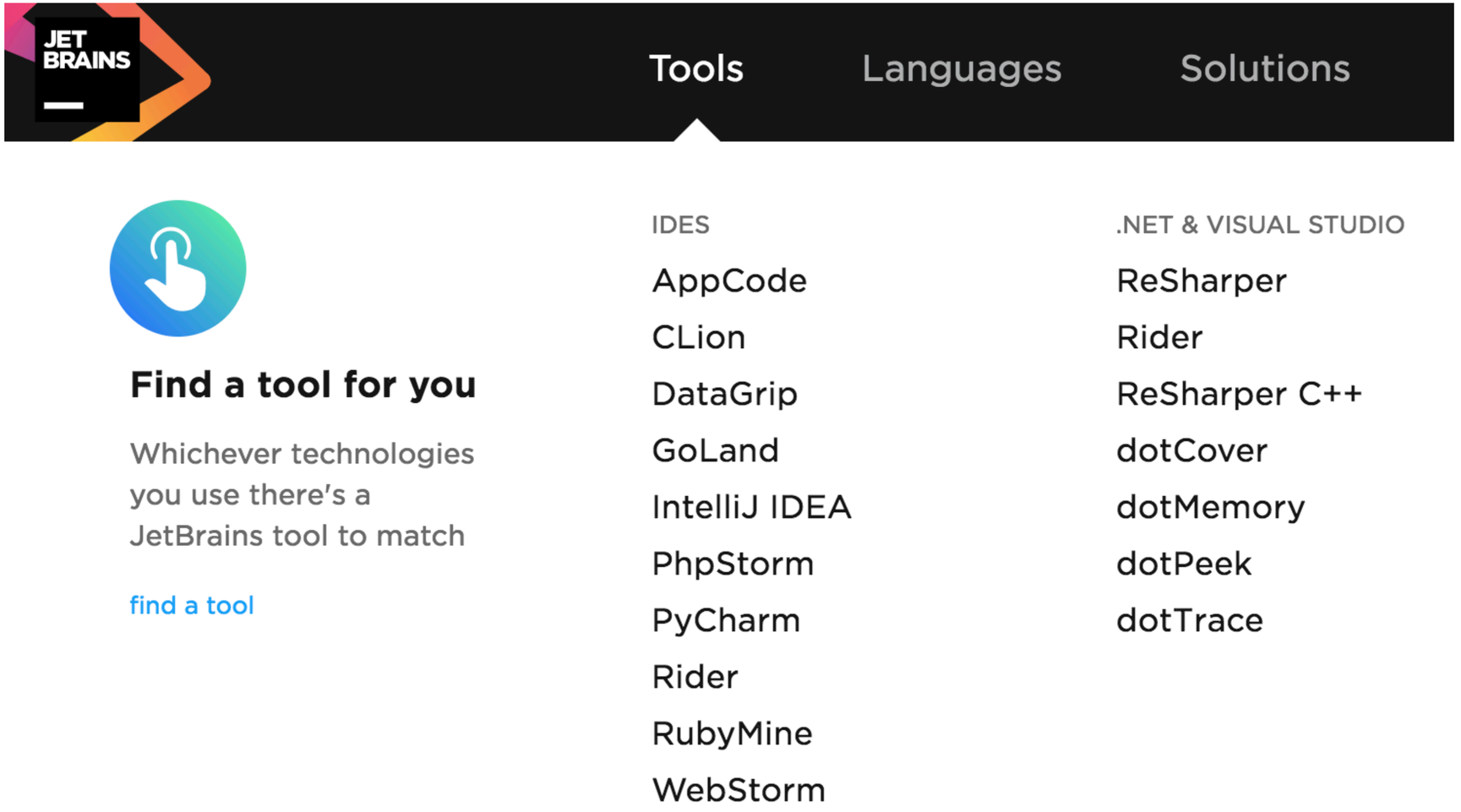
- 使用 Appium Desktop 录制用例
- 安装 Python 依赖 pip install Appium-Python-Client
- 增加隐式等待增强稳定性
- 重新运行

用例结构解析

Appium 客户端安装

Language/Framework	Github Repo and Installation Instructions
Ruby	https://github.com/appium/ruby_lib
Python	https://github.com/appium/python-client
Java	https://github.com/appium/java-client
JavaScript (Node.js)	https://github.com/admc/wd
Objective C	https://github.com/appium/selenium-objective-c
PHP	https://github.com/appium/php-client
C# (.NET)	https://github.com/appium/appium-dotnet-driver
RobotFramework	https://github.com/jollychang/robotframework-appiumlibrary

IDE 环境


JET BRAINS
Tools
Languages
Solutions
Find a tool for you
Whichever technologies you use there's a JetBrains tool to match
find a tool
IDES
AppCode
CLion
DataGrip
GoLand
IntelliJ IDEA
PhpStorm
PyCharm
Rider
RubyMine
WebStorm
.NET & VISUAL STUDIO
ReSharper
Rider
ReSharper C++
dotCover
dotMemory
dotPeek
dotTrace

Java 客户端安装

- IDE 推荐：IntelliJ IDEA
- 配置 Maven 项目
- 在 src/test 下编写用例

```
<dependency>
  <groupId>io.appium</groupId>
  <artifactId>java-client</artifactId>
  <version>${version.you.require}</version>
  <scope>test</scope>
</dependency>
```

Python 客户端安装

- 客户端安装：pip install Appium-Python-Client
- IDE：PyCharm
- Python 多版本隔离工具：venv
- 国内依赖源：http://pypi.douban.com/simple/
- 测试框架：unittest、pytest、nose，推荐 pytest

隐式等待

让用例更健壮的运行起来

- 隐式等待：服务端（Appium）会在特定的超时时间内重试多次寻找控件
- 显式等待：在客户端（用例端）根据更灵活的条件循环等待条件满足

- 在示例代码中加入隐式等待，再次运行

“至此，一个最简单的用例已经跑通，恭喜你入坑”

控件定位方法

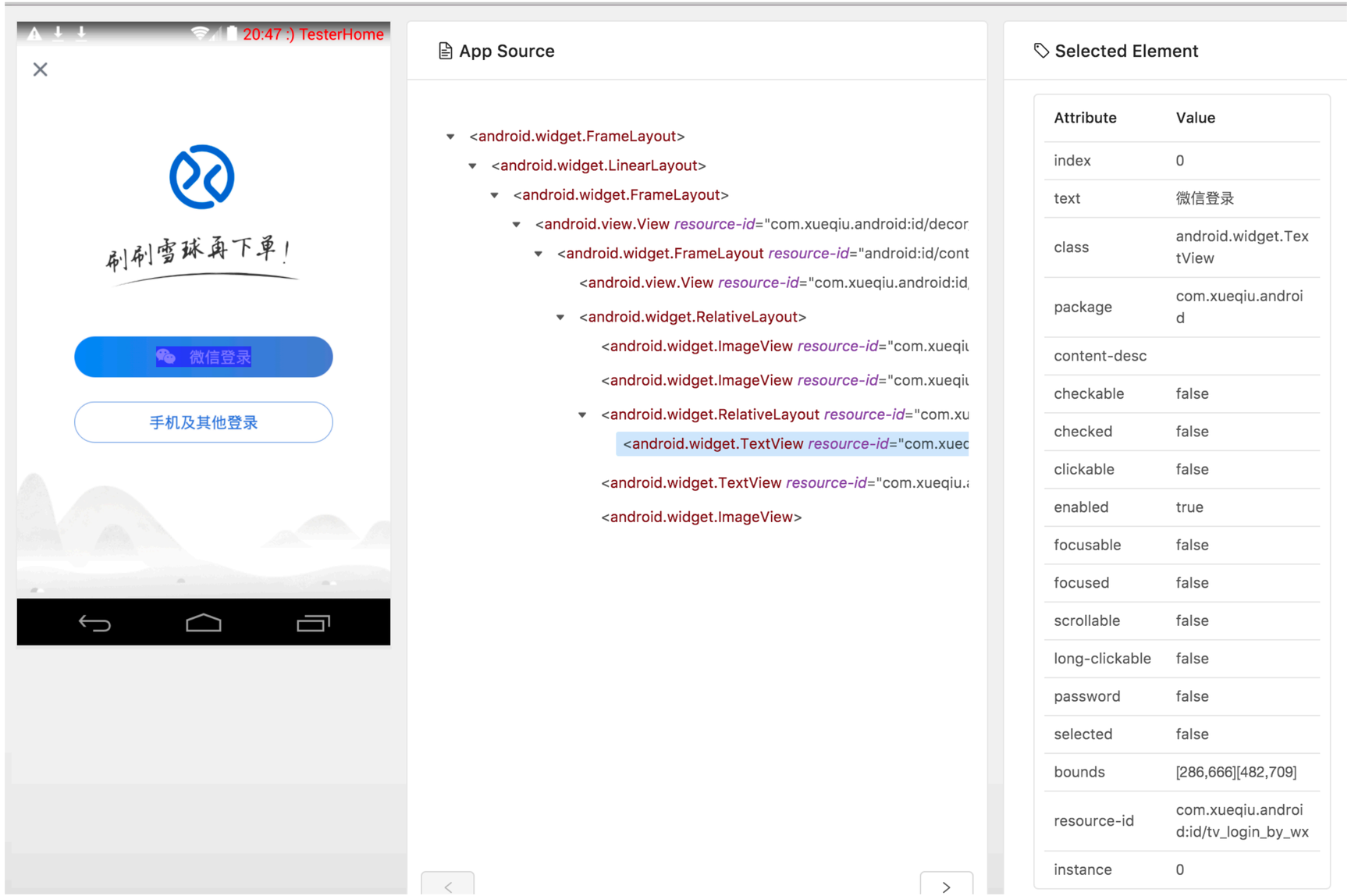
常用定位手段

- id
- Accessibility ID
- XPath

控件基础知识

- DOM：Document Object Model 文档对象模型
- DOM 应用：最早应用于 HTML 和 JavaScript 的交互。界面的结构化描述，常见的格式为 html、xml。核心元素为节点和属性
- XPath：XML 路径语言，用于 XML 中的节点定位

App DOM结构解析


20:47 :) TesterHome
刷刷雪球再下单！
微信登录
手机及其他登录
App Source
<android.widget.FrameLayout>
<android.widget.LinearLayout>
<android.widget.FrameLayout>
<android.view.View resource-id="com.xueqiu.android:id/decor
<android.widget.FrameLayout resource-id="android:id/cont
<android.view.View resource-id="com.xueqiu.android:id
<android.widget.RelativeLayout>
<android.widget.ImageView resource-id="com.xueqi
<android.widget.ImageView resource-id="com.xueqi
<android.widget.RelativeLayout resource-id="com.xu
<android.widget.TextView resource-id="com.xuec
<android.widget.TextView resource-id="com.xueqiu.
<android.widget.ImageView>
Selected Element
Attribute
Value
index
0
text
微信登录
class
android.widget.TextView
package
com.xueqiu.android
content-desc
checkable
false
checked
false
clickable
false
enabled
true
focusable
false
focused
false
scrollable
false
long-clickable
false
password
false
selected
false
bounds
[286,666][482,709]
resource-id
com.xueqiu.android:id/tv_login_by_wx
instance
0

App DOM 为例

- 关键 Attribute
  - clickable
  - content-desc
  - resource-id
  - text
  - bounds
- iOS 与 Android 的区别
  - DOM 属性和节点结构类似
  - 名字和属性的命名不同

```
<android.widget.LinearLayout
  bounds="[198,65][656,131]" checkable="false"
  checked="false"
  class="android.widget.LinearLayout"
  clickable="true" content-desc="" enabled="true"
  focusable="false" focused="false" index="2"
  instance="3" long-clickable="false"
  package="com.xueqiu.android" password="false"
  resource-id="com.xueqiu.android:id/home_search"
  scrollable="false" selected="false" text="">
  <android.widget.ImageView
    bounds="[212,80][248,116]" checkable="false"
    checked="false" class="android.widget.ImageView"
    clickable="false" content-desc="" enabled="true"
    focusable="false" focused="false" index="0"
    instance="3" long-clickable="false"
    package="com.xueqiu.android" password="false"
    resource-id="" scrollable="false"
    selected="false" text=""/>
  <android.widget.TextView
    bounds="[256,79][453,117]" checkable="false"
    checked="false" class="android.widget.TextView"
    clickable="false" content-desc="" enabled="true"
    focusable="false" focused="false" index="1"
    instance="0" long-clickable="false"
    package="com.xueqiu.android" password="false"
    resource-id="com.xueqiu.android:id/tv_search"
    scrollable="false" selected="false" text="雪球Club南京站"
</android.widget.LinearLayout>
<android.widget.FrameLayout NAF="true"
  bounds="[680,50][744,146]" checkable="false"
  checked="false" class="android.widget.FrameLayout"
  clickable="true" content-desc="" enabled="true"
```

元素定位

- 测试步骤三要素：
  - 定位、交互、断言
- 定位
  - ID（重要）
  - XPath（重要）
  - Accessibility ID: content-desc（重要）
  - 不推荐：Class -iOS -Android

```
private Object findElement(By by) throws InvalidSelectorException, ElementNotF
    if (by instanceof ById) {
        String locator = getElementLocator((ById)by);
        return getInstance().findObject(android.support.test.uiautomator.By.re
    } else if (by instanceof By.ByAccessibilityId) {
        return getInstance().findObject(android.support.test.uiautomator.By.de
    } else if (by instanceof ByClass) {
        return getInstance().findObject(android.support.test.uiautomator.By.cl
    } else if (by instanceof By.ByXPath) {
        return getXPathUiObject(by.getElementLocator(), null /* AndroidElement
    } else if (by instanceof By.ByAndroidUiAutomator) {
        return getInstance().findObject(findByUiAutomator(by.getElementLocator
    }
    String msg = String.format("By locator %s is currently not supported!", by
    throw new UnsupportedOperationException(msg);
}
```

UI Automator 2 的定位逻辑

```
private Object findElement(By by) throws InvalidSelectorException, ElementNotFoundException, ParserConfigurationException,
    if (by instanceof ById) {
        String locator = getElementLocator((ById)by);
        return getInstance().findObject(android.support.test.uiautomator.By.res(locator));
    } else if (by instanceof By.ByAccessibilityId) {
        return getInstance().findObject(android.support.test.uiautomator.By.desc(by.getElementLocator()));
    } else if (by instanceof ByClass) {
        return getInstance().findObject(android.support.test.uiautomator.By.clazz(by.getElementLocator()));
    } else if (by instanceof By.ByXPath) {
        return getXPathUiObject(by.getElementLocator(), null /* AndroidElement */);
    } else if (by instanceof By.ByAndroidUiAutomator) {
        return getInstance().findObject(findByUiAutomator(by.getElementLocator()));
    }
    String msg = String.format("By locator %s is currently not supported!", by.getClass().getSimpleName());
    throw new UnsupportedOperationException(msg);
}
```

定位与操作的代码示例

```python
def test_simple_actions(self):
    el = self.driver.find_element_by_accessibility_id('Graphics')
    el.click()

    el = self.driver.find_element_by_accessibility_id('Arcs')
    el.click()

    self.driver.find_element_by_android_uiautomator('new UiSelector().text("Graphics/Arcs")')
```

```java
@Test
public void apiDemo(){
    WebElement el = driver.findElement(By.xpath(".//*[@text='Animation']"));
    assertEquals("Animation", el.getText());
    el = driver.findElementByClassName("android.widget.TextView");
    assertEquals("API Demos", el.getText());
    el = driver.findElement(By.xpath(".//*[@text='App']"));
    el.click();
    List<WebElement> els = driver.findElementsByClassName("android.widget.TextView");
    assertEquals("Activity", els.get(2).getText());
}
```

元素定位符与复用

- findElementByXXX
- findElement(by, value)

- findElement 主要用于 Page Object 模式

常用自动化 API

常见自动化动作支持

- click
- sendKeys
- swipe
- touch action

```python
# python
driver.swipe(start_x=75, start_y=500, end_
```

```java
// java
driver.swipe(75, 500, 75, 0, 0.8)
```

TouchAction 应用

手势操作 TouchAction

- press release longPress
- tap wait
- moveTo
- perform

Java | Python | Javascript | Ruby | C# | PHP

```python
from appium.webdriver.common.touch_action import TouchAction
// ...
actions = TouchAction(driver)
actions.tap_and_hold(20, 20)
actions.move_to(10, 100)
actions.release()
actions.perform()
```

capability 使用进阶

capabilities 设置

- App APK 地址 appPackage 包名 appActivity Activity 名字
- automationName 默认使用 uiautomator
- noReset fullReset 是否在测试前后重置相关环境
- autoGrantPermissions 自动赋予 App 权限
- unicodeKeyBoard resetKeyBoard 是否需要输入非英文之外的语言并在测试完成后重置输入法

- 更多参考官网文档

capabilities 示例代码

```python
# Android environment
import unittest
from appium import webdriver

desired_caps = {}
desired_caps['platformName'] = 'Android'
desired_caps['platformVersion'] = '4.2'
desired_caps['deviceName'] = 'Android Emulator'
desired_caps['app'] = PATH('../../../apps/selendroid-test-app.apk')

self.driver = webdriver.Remote('http://localhost:4723/wd/hub', desired_caps)
```

```java
import java.io.File;
import org.openqa.selenium.remote.DesiredCapabilities;
import io.appium.java_client.AppiumDriver;
import io.appium.java_client.android.AndroidDriver;
import io.appium.java_client.MobileElement;
import java.net.URL;

...
File app  = new File("The absolute or relative path to an *.apk file");
DesiredCapabilities capabilities = new DesiredCapabilities();
capabilities.setCapability(MobileCapabilityType.DEVICE_NAME, "Android Emulator");
capabilities.setCapability(MobileCapabilityType.APP, app.getAbsolutePath());
capabilities.setCapability(MobileCapabilityType.PLATFORM_NAME, MobilePlatform.ANDROID);
//you are free to set additional capabilities
AppiumDriver<MobileElement> driver = new AppiumDriver<>(
new URL("http://target_ip:used_port/wd/hub"), //if it needs to use locally started server
//then the target_ip is 127.0.0.1 or 0.0.0.0
//the default port is 4723
capabilities);
```

极客时间

扫码试看/订阅

《移动端自动化测试实战》视频课程