

视频笔记：

- iOS 测试
 - iOS 平台的封闭性
 - 主流移动测试框架
 - Appium、Calabash-iOS、KIF、XCTest（苹果）、WebDriverAgent、UiAutomation
 - 证书体系
- Appium 启动 app
 - Register iOS App IDs、Bundle ID、Enable Services 等注册信息
 - iOS Provisioning Profiles
 - 发布方式：Ad-hoc、In House、App Store
 - 常见命令：
 - 依赖工具包安装 brew install libimobiledevice
 - 查看模拟器列表 instruments -s devices
 - 查看真机列表 idevices_id -l
 - 安装 app ideviceinstaller -i demo.app
 - 命令行编译 xcodebuild -scheme UICatalog -target iOS clean build
 - xcodebuild -scheme UICatalog -target iOS archive
 - 元素定位方式：name、label、value
 - 例子：
 - 在左侧最后一个图标，UICatalog（App 名字）中 Build 成功后，找到绿色箭头 Sign UICatalog.app 下 Signing Identity 内.app 的路径地址。
 - 打开 Appium Desktop，输入 platformName 为 ios、deviceName 为 iPhone 8、platformVersion 为 11.2、app 为上一步中的路径地址、automationName 为 xcuitest。启动模拟器。尽量新建一个端口。
- 使用 Inspector 定位
 - 安装依赖，运行 WebDriverAgent 通过输入 ./Scripts/bootstrap.sh。参照 <https://testerhome.com/topics/9666>
 - 使用 Inspector 主要参照 <https://github.com/facebook/WebDriverAgent/wiki/Using-the-Inspector>
 - 设置 dev tools 和 deps：
 - DevToolsSecurity --enable
 - carthage bootstrap --platform ios
 - Inspector GUI 图形化：http://localhost:8100/inspector
 - JSON elements tree：http://localhost:8100/source
 - 构建方法：xcodebuild -project WebDriverAgent.xcodeproj \ -scheme WebDriverAgentRunner \ -destination 'platform=iOS Simulator,name=iPhone 8' \ test
 - 在 Inspector 中元素一般为<XCUIElementTypeCell>、<XCUIElementTypeStaticText>等。
 - 如某个<XCUIElementTypeButton>的 value、name、label 都为 Buttons，而某个<XCUIElementTypeTextField>的 value 是 Placeholder text，即其输入框内部默认值，label 为空，没 name 属性。
- iOS 用例演练
 - 以 Java 代码形式完成 appium 的初始化操作，并完成简单的用例。

```
import org.junit.Test;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;

import java.net.MalformedURLException;
import java.net.URL;

public class TestIOS {
    @Test
    public void UICatalogTest() throws MalformedURLException {
        DesiredCapabilities caps=new DesiredCapabilities();
        caps.setCapability("platformName", "ios");
        caps.setCapability("platformVersion", "11.2");
        caps.setCapability("deviceName", "iPhone 8");
        caps.setCapability("app",
            "/Users/seveniruby/Library/Developer/Xcode/DerivedData/UICatalog-ftyzdbgapijmxobezrnrnxssh
            "/Products/Debug-iphonesimulator/UICatalog.app");

        RemoteWebDriver driver=new RemoteWebDriver(new URL( spec: "http://127.0.0.1:4723/wd/hub"), caps);
        System.out.println(driver.getPageSource());
        for(WebElement e: driver.findElementsByXPath( using: "//*")){
            System.out.println(e.getText());
            System.out.println(e.getTagName());
        }
    }

    @Test
    public void UICatalogTestClick() throws MalformedURLException {
        DesiredCapabilities caps=new DesiredCapabilities();
        caps.setCapability( capabilityName: "platformName", value: "ios");
        caps.setCapability( capabilityName: "platformVersion", value: "11.2");
        caps.setCapability( capabilityName: "deviceName", value: "iPhone X");
        caps.setCapability( capabilityName: "app",
            value: "/Users/seveniruby/Library/Developer/Xcode/DerivedData/UICatalog-ftyzdbgapijmxobezrnrnxssh
            "/Products/Debug-iphonesimulator/UICatalog.app");

        AppiumDriver driver=new AppiumDriver(new URL( spec: "http://127.0.0.1:4723/wd/hub"), caps);
        driver.manage().timeouts().implicitlyWait( time: 10, TimeUnit.SECONDS);
        System.out.println(driver.getPageSource());
        driver.findElementByAccessibilityId( using: "Buttons").click();
        System.out.println(driver.findElementsByXPath( using: "//*[contains(@type, 'Button')]"));
    }
}
```