iOS-XCUITest使用教程

前提：测试的App基于iOS 9.3 以及之后的系统版本，Xcode软件版本不低于7，才可以使用该技术，UI Testing 在易用性上比 KIF 这样的框架要有所进步，随着 UI Testing 的推出，Apple 也将原来的 UI Automation 一系列内容标记为弃用。这意味着 UI Testing 至少在今后一段时间内将会是 iOS 开发中的首选工具。但是我们也应该看到，基于 Accessibility 的测试方式有时候并不是很直接。在这个限制下，我们只能得到 UI 的代理对象，而不是 UI 元素本身，这让我们无法得到关于 UI 元素更多的信息 (比如直接获取 UI 元素中的内容，或者与 ViewController 中的相关的值)，现在的 UI Testing 在很大程度上还停留在比较简易的阶段，适合保证App的主流程的顺畅运行、用于回归测试。在技术探索过程中，发现测试脚本在iOS 11系统表现很好，但是在iOS 9等系统上会出现找不到按钮的情况等问题。随着iOS系统版本的推进，相信XCUITest技术的兼容性、性能会不断得到改进。

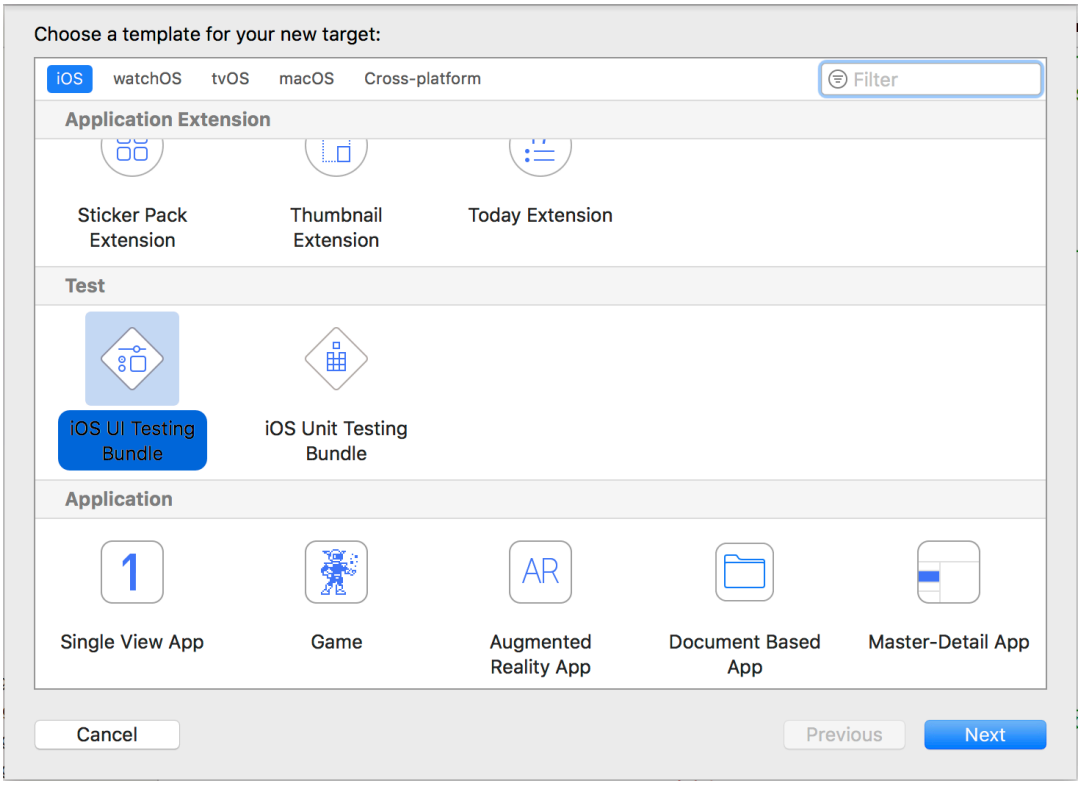
一、给已有的Xcode源码工程添加UITest功能：

1.在项目中新建UITest模块

（1）如果是新建空白项目，可以勾选Include UITest相关选项。

（2）如果是已有项目（比如pull下来的完整的iOS客户端代码），在Xcode中点击File->New->Target

如图选择：



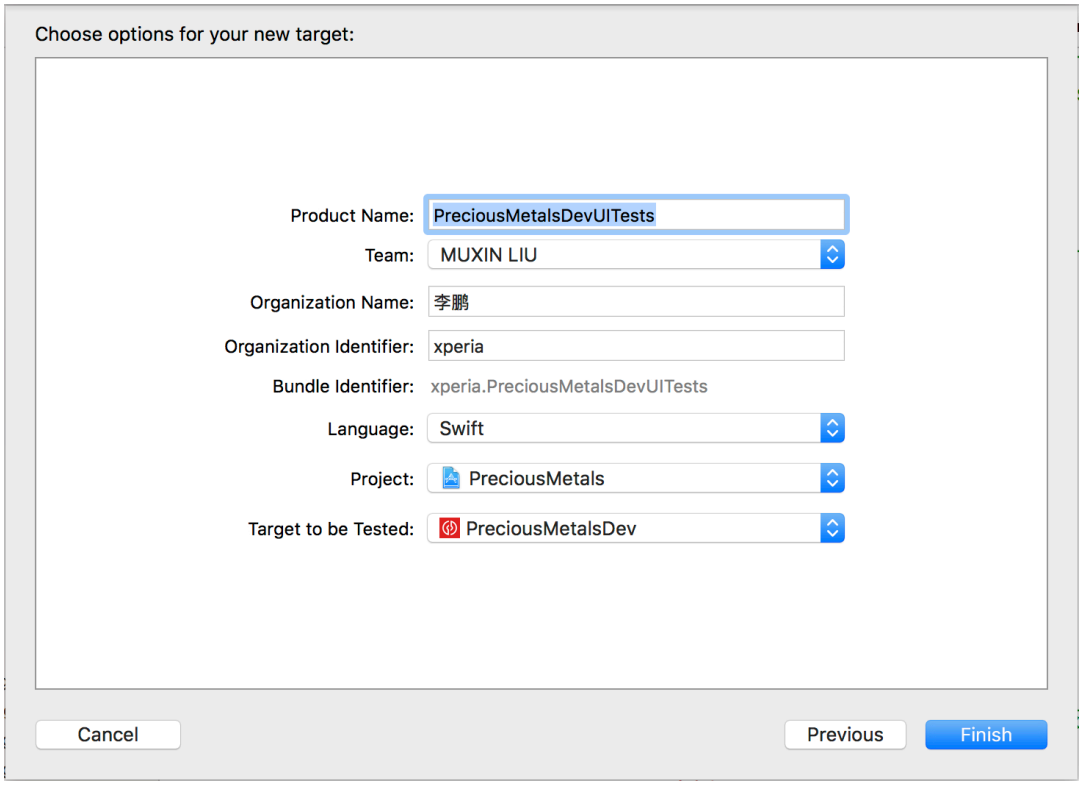
点击Next，进入选项配置：

第一步：选择Project。

第二步：选择最底部的Target to be Tested，这里选择Dev结尾的包，因为该包在模拟器上测试不需要签名。

此时Product Name会根据选项自动生成。切换自己需要的语言（Objective-C、Swift（推荐））

完成。



此时，项目中增加了新的以[Target+UITest]命名的文件夹，如图：

当前选中的与文件夹同名的（UITests结尾）文件就是我们可以编写测试代码的文件了。

2.测试代码文件介绍

在这里有初始化的几个方法 ：

setup：程序启动时会进入该方法，可以设置一些全局属性，比测试如用例失败是继续执行还是停止，是否需要禁用App的过渡动画，启动的时候的屏幕方向等。

tearDown：程序结束时会进入该方法。

tearExmple：测试方法，该方法系统生成，用于示例，我们可以自己写一些其他的方法，但是需要test打头，这样系统会识别出该方法为测试方法。

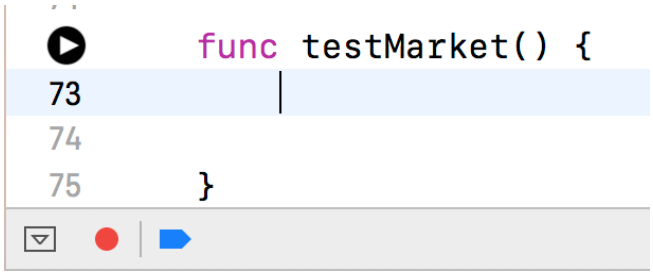
执行代码的方法：

当鼠标移动到tearExmple函数时，会发现这一行前面出现一个播放形状(▶)的小图标，点击该图标程序会执行该测试方法。当遇到问题，或者没有达到预期值得时候，方法后面会出现红色，方法通过之后会出现绿色。

快捷键：control+option+command+U，当光标放在某个测试函数的代码中时，同样有效。

二、代码录制功能

在编写之前，了解下Xcode给提供的代码录制功能。



在上图中，我们找到控制台输出，哪里有一个红色的小圆点，先别着急点。先我们的光标移动到testMarket方法里之后，点击红色小圆点。进入录制阶段，等待程序的启动。当程序启动之后，点击程序的一个按钮，在光标相应的地方会生成一段代码。

三、常用基本操作

1.点击某个按钮，比如登录按钮

```
app.buttons["登录"].tap()
```

2.普通输入框输入文本

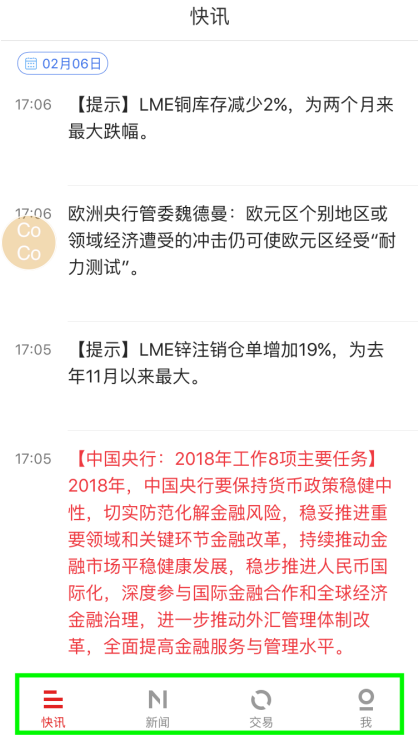
```
app.textFields["手机号"].tap()                    //要先聚焦文本框，才能继续输
```

入
app.textFields["手机号"].typeText("13038865629")
注意密码输入框等输入文本会变成小圆点的地方使用app.secureTextFields["密码"]。

3.下拉刷新
app.swipeDown()

4.滚动查看屏幕下方的内容，上拉加载更多
app.swipeUp()
也可以左右滑动切换页面：app.swipeLeft/Right()

5.切换Tab



app.tabBars.buttons["首页"].tap()
app.tabBars.buttons["行情"].tap()

6.让应用程序等待一会（比如数据载入的时候，或者延时操作）
sleep(3)　　//单位是秒

7.判断元素是否存在
if app.buttons["确定"].exists {

```
    //点击确定按钮
}
```

8.使用断言来验证App界面的状态

XCTAssert（app.buttons["个人中心"].count == 1）　　//count属性表示符合查询条件的结果个数

9.从导航栏返回上级页面

app.navigationBars["消息中心"].buttons["返回"].tap()

10.使用正则匹配，操作列表元素（进阶）


```
let addButton =  app.tables.cells.matching(NSPredicate(format: "label CONTAINS %@", "铜")).element(boundBy: 0)
addButton.tap()
//NSPredicate(format: "label CONTAINS %@", "铜") -- 筛选包含‘铜’字的文本
//element(boundBy: 0) -- 查询结果的第一个元素，该页面结果显然是唯一的，这种情况下可以使用firstMatch属性来代替，提升效率。
```

四、通过命令行执行UI自动化测试

做平台集成需要了解的知识

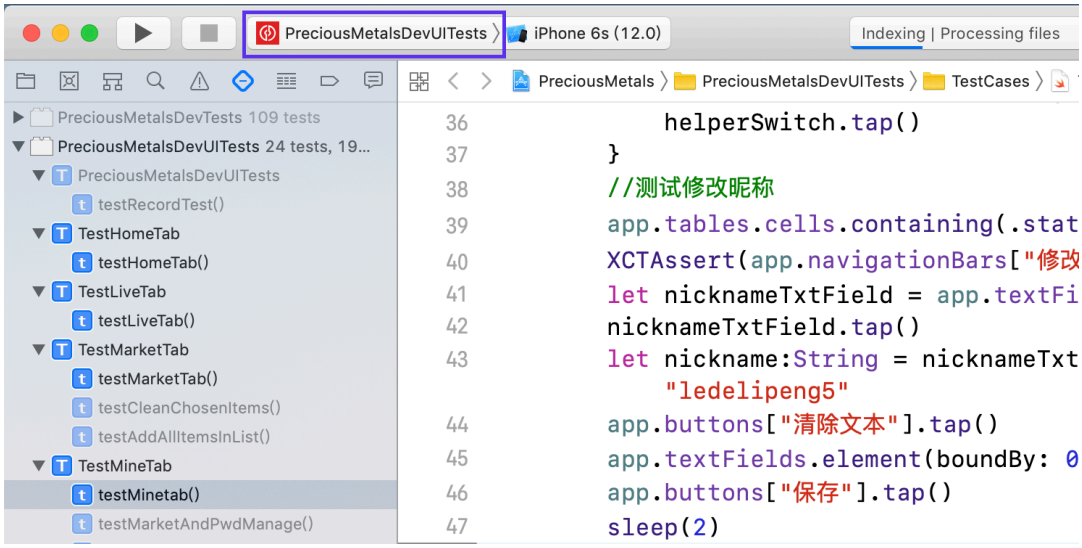
首先进入项目中包含PreciousMetals.xcproject文件的文件夹下，然后执行：

```
xcodebuild test -workspace PreciousMetals.xcworkspace -scheme PreciousMetalsDevUITests -destination 'platform=iOS Simulator,name=iPhone X,OS=12.0'
```

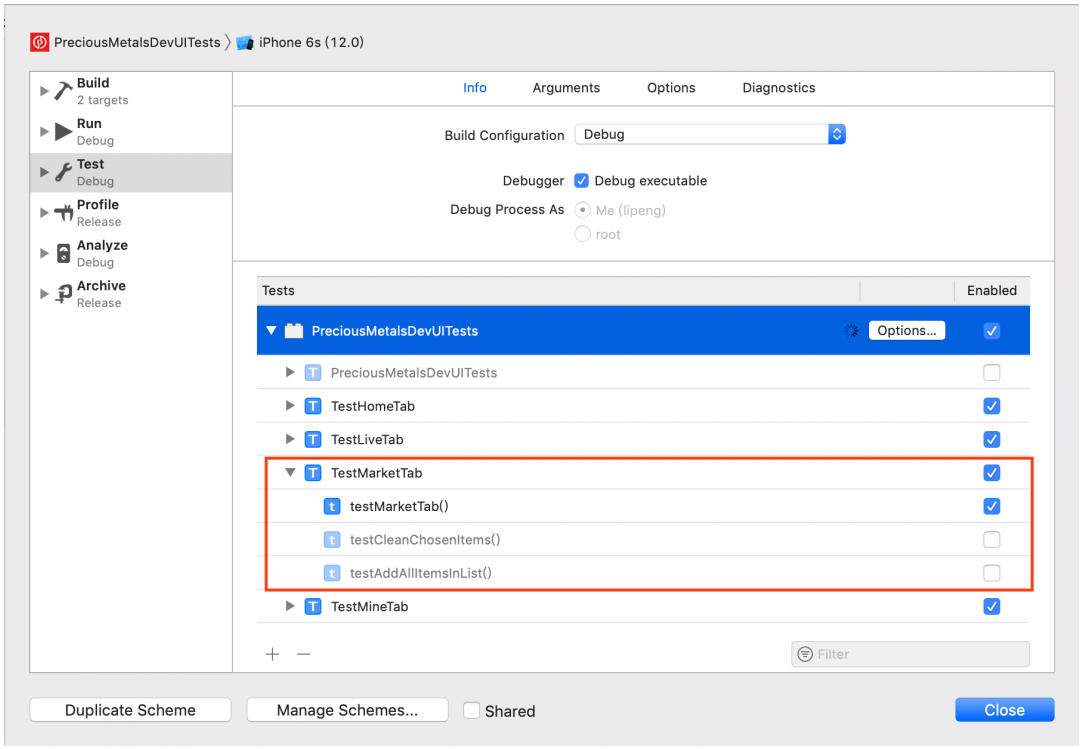
-workspace参数： 项目的工程文件，和使用xcode打开客户端项目时双击的文件是同一个。

-scheme参数：使用的配置或者方案，就是测试哪个目标，测试过程有什么代码覆盖等要求都可以在scheme中配置。

-destination参数可以指定多个，从而实现一个命令执行多个模拟器的兼容测试。



编辑 PreciousMetalsDevUITests 测试计划，可以配置以test开头命名的各个函数是否需要测试。



五、测试报告的优化

1.优化测试报告的可读性（层级关系）,同时改进了源码可读性。

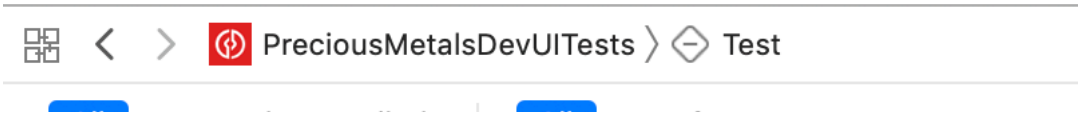
```
XCTContext.runActivity(named: "测试直播间内 - 子Tab切换", block: { _ in
        app.otherElements["策略"].tap()
        app.otherElements["提问"].tap()
```

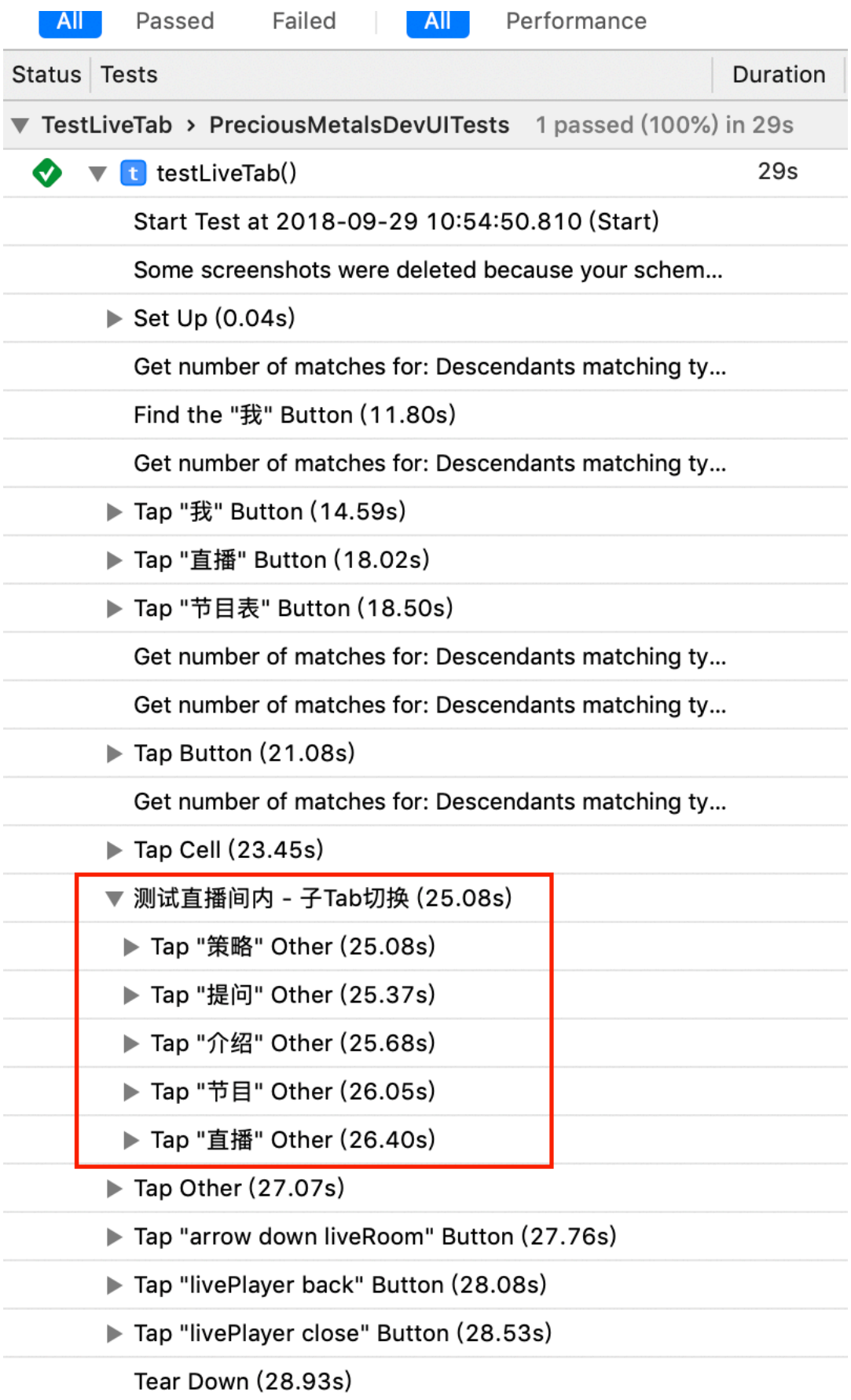
```
        app.otherElements["介绍"].tap()
        app.otherElements["节目"].tap()
        app.otherElements["直播"].tap()
    })
```

效果图：控制台输入，即为

```
t =    18.02s Tap "直播" Button
t =    18.02s     Wait for com.netease.gold.dev to idle
t =    18.06s     Find the "直播" Button
t =    18.19s         Check for interrupting elements affecting "直播" Button
t =    18.19s     Synthesize event
t =    18.30s     Wait for com.netease.gold.dev to idle
t =    18.50s Tap "节目表" Button
t =    18.50s     Wait for com.netease.gold.dev to idle
t =    18.60s     Find the "节目表" Button
t =    18.69s         Check for interrupting elements affecting "节目表" Button
t =    18.70s     Synthesize event
t =    18.82s     Wait for com.netease.gold.dev to idle
t =    20.96s Get number of matches for: Descendants matching type Button
t =    21.02s Get number of matches for: Descendants matching type Button
t =    21.08s Tap Button
t =    21.08s     Wait for com.netease.gold.dev to idle
t =    21.13s     Find the Button
t =    21.16s         Check for interrupting elements affecting "直播" Button
t =    21.16s     Synthesize event
t =    21.27s     Wait for com.netease.gold.dev to idle
t =    21.33s Get number of matches for: Descendants matching type Cell
t =    23.45s Tap Cell
t =    23.45s     Wait for com.netease.gold.dev to idle
t =    23.52s     Find the Cell
t =    23.61s         Check for interrupting elements affecting Cell
t =    23.61s     Synthesize event
t =    23.74s     Wait for com.netease.gold.dev to idle
t =    25.08s 测试直播间内 - 子Tab切换
t =    25.08s     Tap "策略" Other
t =    25.08s         Wait for com.netease.gold.dev to idle
t =    25.14s         Find the "策略" Other
t =    25.19s             Check for interrupting elements affecting "策略" Other
t =    25.20s         Synthesize event
t =    25.31s         Wait for com.netease.gold.dev to idle
t =    25.37s     Tap "提问" Other
t =    25.37s         Wait for com.netease.gold.dev to idle
t =    25.44s         Find the "提问" Other
t =    25.49s             Check for interrupting elements affecting "提问" Other
t =    25.50s         Synthesize event
t =    25.62s         Wait for com.netease.gold.dev to idle
t =    25.69s     Tap "介绍" Other
t =    25.69s         Wait for com.netease.gold.dev to idle
t =    25.80s         Find the "介绍" Other
t =    25.87s             Check for interrupting elements affecting "介绍" Other
t =    25.87s         Synthesize event
t =    25.99s         Wait for com.netease.gold.dev to idle
t =    26.05s     Tap "节目" Other
t =    26.05s         Wait for com.netease.gold.dev to idle
t =    26.13s         Find the "节目" Other
t =    26.19s             Check for interrupting elements affecting "节目" Other
t =    26.20s         Synthesize event
t =    26.33s         Wait for com.netease.gold.dev to idle
```

Xcode内置的测试报告预览界面，可以快速定位测试结果中的问题。




All
Passed
Failed
All
Performance
Status
Tests
Duration
TestLiveTab › PreciousMetalsDevUITests 1 passed (100%) in 29s
testLiveTab() 29s
Start Test at 2018-09-29 10:54:50.810 (Start)
Some screenshots were deleted because your schem...
Set Up (0.04s)
Get number of matches for: Descendants matching ty...
Find the "我" Button (11.80s)
Get number of matches for: Descendants matching ty...
Tap "我" Button (14.59s)
Tap "直播" Button (18.02s)
Tap "节目表" Button (18.50s)
Get number of matches for: Descendants matching ty...
Get number of matches for: Descendants matching ty...
Tap Button (21.08s)
Get number of matches for: Descendants matching ty...
Tap Cell (23.45s)
测试直播间内 - 子Tab切换 (25.08s)
Tap "策略" Other (25.08s)
Tap "提问" Other (25.37s)
Tap "介绍" Other (25.68s)
Tap "节目" Other (26.05s)
Tap "直播" Other (26.40s)
Tap Other (27.07s)
Tap "arrow down liveRoom" Button (27.76s)
Tap "livePlayer back" Button (28.08s)
Tap "livePlayer close" Button (28.53s)
Tear Down (28.93s)

2.增加指定元素的截图到结果集：

```
XCTContext.runActivity(named:"收集一波关键截图") {  activity   in
    //捕捉第一个cell的截图
    //也可以使用XCUIScreen.main.screenshot()来捕捉一张全屏幕截图
    let  cell =  app.cells.element(boundBy:0)
    let cellAttachment =  XCTAttachment(screenshot:  cell.screenshot())
    cellAttachment.lifetime =  .keepAlways
    activity.add(cellAttachment)
}
```