

# 目录

前言	1.1
概述	1.2
通用	1.3
桌面端	1.4
XCode	1.4.1
xcodebuild	1.4.1.1
xcrun	1.4.1.2
ideviceinstaller	1.4.2
端口转发	1.4.3
idevice_id	1.4.4
system_profiler	1.4.5
instruments	1.4.6
security	1.4.7
移动端	1.5
iOS	1.5.1
iPhone	1.5.1.1
附录	1.6
文档	1.6.1
参考资料	1.6.2

# 苹果相关开发总结

- 最新版本: v1.1
- 更新时间: 20200625

## 简介

总结Apple苹果相关领域的开发经验和心得，包括常见的开发工具，比如XCode、xcodebuild、xcrun、ideviceinstaller、iproxy或mobiledevice的端口转发、libimobiledevice的idevice\_id、system\_profiler、instruments、security；以及移动端的iOS系统，比如自动化框架XCTest、官网文档简介、微信使用心得、真机iPhone开发心得；最后给出相关文档和参考资料。

## 源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

### Gitbook源码

- [crifan/apple\\_develop\\_summary](#): 苹果相关开发总结

如何使用此Gitbook源码去生成发布为电子书

详见：[crifan/gitbook\\_template](#): demo how to use crifan gitbook template and demo

### 在线浏览

- 苹果相关开发总结 [book.crifan.com](#)
- 苹果相关开发总结 [crifan.github.io](#)

### 离线下载阅读

- [苹果相关开发总结 PDF](#)
- [苹果相关开发总结 ePUB](#)
- [苹果相关开发总结 Mobi](#)

### 版权说明

此电子书教程的全部内容，如无特别说明，均为本人原创和整理。其中部分内容参考自网络，均已备注了出处。如有发现侵犯您版权，请通过邮箱联系我 `admin 艾特 crifan.com`，我会尽快删除。谢谢合作。

### 鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 crifan 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

## 更多其他电子书

本人 crifan 还写了其他 100+ 本电子书教程，感兴趣可移步至：

[crifan/crifan\\_ebook\\_readme: Crifan的电子书的使用说明](#)

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-01-16 23:56:39

# 苹果相关开发概述

此处主要整理苹果Apple相关开发资料和开发工具

- 桌面端
  - Mac = macOS
- 移动端
  - 系统
    - iOS
  - 设备
    - iPhone
  - 开发语言
    - Swift
    - Objective-C

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:23:59

## 通用

此处介绍苹果开发涉及到的通用的内容。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:23:59

## 桌面端

苹果开发中桌面端，主要指的是：

- 物理设备：Mac电脑
  - 包括 Mac Pro 、 Mac Air 等
- 操作系统：
  - 最新叫：macOS
  - 之前叫：OS X

接下来介绍Mac中各种有用的开发工具。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:23:59

## XCode

XCode是Mac中用来开发应用的IDE。

支持各种Apple的系统：

- 桌面端：`macOS`
- 移动端：
  - iPhone类：`iOS`
  - iPad类：`ipadOS`
  - Apple TV：`tvOS`
  - Apple Watch：`watchOS`

XCode作为一个IDE之外，还内置附带（和与之相关）了其他一些命令行工具，下面整理如下。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:23:59

## xcodebuild

- `xcodebuild`
  - 是什么：XCode的命令行工具
  - 作用：用于命令行方式去编译项目
  - 好处：而无需打开XCode图形界面去操作
    - 可用于自动化部署等方面

用法举例：

```
UDID ed94089f3e34d5538065a695bfd03dfbb3c5579
xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination "id=$UDID" test
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:23:59

## xcrun

xcrun simctl list devices

```
xcrun simctl list devices
Devices
-- iOS 13.3 --
iPhone 8 (54589698-0C9F-407D-B21A-83432CABB681) (Shutdown)
iPhone 8 Plus (509B7103-97DB-4AB9-B829-001190ED4B7E) (Shutdown)
iPhone 11 (509BC7C7-9C0E-42FA-8AB2-F5220EBAA13B) (Shutdown)
iPhone 11 Pro (3E8E7E92-66F2-4AF3-A405-23B5FB231DE7) (Shutdown)
iPhone 11 Pro Max (50C15135-1532-44C5-B82C-B327F88F2712) (Shutdown)
iPad Pro (9.7-inch) (B11D5D40-FEA2-4114-B053-E4CFD29D127C) (Shutdown)
iPad (7th generation) (7F8EDE89-74E0-4BAB-B3CA-09E2DAE1F095) (Shutdown)
iPad Pro (11-inch) (04DD3B8A-5B78-48E8-8B22-56796A9CFB73) (Shutdown)
iPad Pro (12.9-inch) (3rd generation) (D811684E-2F3E-4FC6-92EA-39301451F7E5) (Shutdown)

iPad Air (3rd generation) (BBC48526-3922-4C97-BA14-B1888385243A) (Shutdown)
-- tvOS 13.3 --
Apple TV (6680F059-4DE1-430C-B696-228AC27CAA88) (Shutdown)
Apple TV 4K (048E58E8-6A27-4D81-BDEB-8812C610B756) (Shutdown)
Apple TV 4K (at 1080p) (384D5E60-B6B1-481E-BDC3-B7FF8F773412) (Shutdown)
-- watchOS 6.1 --
Apple Watch Series 4 - 40mm (1B98415B-3FDE-401B-A80C-A3551DB207D7) (Shutdown)
Apple Watch Series 4 - 44mm (661838E9-B0BE-42B4-B55E-9A34263B1AEA) (Shutdown)
Apple Watch Series 5 - 40mm (F76D77EF-0932-4164-94BB-9FC757420911) (Shutdown)
Apple Watch Series 5 - 44mm (D86F0BD5-4D38-4537-9C8C-2F5C74E404CA) (Shutdown)
```

xcrun simctl list

```
xcrun simctl list
== Device Types ==
iPhone 4s (com.apple.CoreSimulator.SimDeviceType.iPhone-4s)
iPhone 5 (com.apple.CoreSimulator.SimDeviceType.iPhone-5)
iPhone 5s (com.apple.CoreSimulator.SimDeviceType.iPhone-5s)
iPhone 6 Plus (com.apple.CoreSimulator.SimDeviceType.iPhone-6-Plus)
iPhone 6 (com.apple.CoreSimulator.SimDeviceType.iPhone-6)
iPhone 6s (com.apple.CoreSimulator.SimDeviceType.iPhone-6s)
iPhone 6s Plus (com.apple.CoreSimulator.SimDeviceType.iPhone-6s-Plus)
iPhone SE (com.apple.CoreSimulator.SimDeviceType.iPhone-SE)
iPhone 7 (com.apple.CoreSimulator.SimDeviceType.iPhone-7)
iPhone 7 Plus (com.apple.CoreSimulator.SimDeviceType.iPhone-7-Plus)
iPhone 8 (com.apple.CoreSimulator.SimDeviceType.iPhone-8)
iPhone 8 Plus (com.apple.CoreSimulator.SimDeviceType.iPhone-8-Plus)
iPhone X (com.apple.CoreSimulator.SimDeviceType.iPhone-X)
iPhone Xs (com.apple.CoreSimulator.SimDeviceType.iPhone-XS)
iPhone Xs Max (com.apple.CoreSimulator.SimDeviceType.iPhone-XS-Max)
```

```
iPhone XR (com.apple.CoreSimulator.SimDeviceType.iPhone-XR)
iPhone 11 (com.apple.CoreSimulator.SimDeviceType.iPhone-11)
iPhone 11 Pro (com.apple.CoreSimulator.SimDeviceType.iPhone-11-Pro)
iPhone 11 Pro Max (com.apple.CoreSimulator.SimDeviceType.iPhone-11-Pro-Max)
iPad 2 (com.apple.CoreSimulator.SimDeviceType.iPad-2)
iPad Retina (com.apple.CoreSimulator.SimDeviceType.iPad-Retina)
iPad Air (com.apple.CoreSimulator.SimDeviceType.iPad-Air)
iPad mini 2 (com.apple.CoreSimulator.SimDeviceType.iPad-mini-2)
iPad mini 3 (com.apple.CoreSimulator.SimDeviceType.iPad-mini-3)
iPad mini 4 (com.apple.CoreSimulator.SimDeviceType.iPad-mini-4)
iPad Air 2 (com.apple.CoreSimulator.SimDeviceType.iPad-Air-2)
iPad Pro (9.7-inch) (com.apple.CoreSimulator.SimDeviceType.iPad-Pro--9-7-inch-)
iPad Pro (12.9-inch) (com.apple.CoreSimulator.SimDeviceType.iPad-Pro)
iPad (5th generation) (com.apple.CoreSimulator.SimDeviceType.iPad--5th-generation-)
iPad Pro (12.9-inch) (2nd generation) (com.apple.CoreSimulator.SimDeviceType.iPad-Pro--12-9-inch---2nd-generation-)
iPad Pro (10.5-inch) (com.apple.CoreSimulator.SimDeviceType.iPad-Pro--10-5-inch-)
iPad (6th generation) (com.apple.CoreSimulator.SimDeviceType.iPad--6th-generation-)
iPad (7th generation) (com.apple.CoreSimulator.SimDeviceType.iPad--7th-generation-)
iPad Pro (11-inch) (com.apple.CoreSimulator.SimDeviceType.iPad-Pro--11-inch-)
iPad Pro (12.9-inch) (3rd generation) (com.apple.CoreSimulator.SimDeviceType.iPad-Pro--12-9-inch---3rd-generation-)
iPad mini (5th generation) (com.apple.CoreSimulator.SimDeviceType.iPad-mini--5th-generation-)
iPad Air (3rd generation) (com.apple.CoreSimulator.SimDeviceType.iPad-Air--3rd-generation-)

Apple TV (com.apple.CoreSimulator.SimDeviceType.Apple-TV-1080p)
Apple TV 4K (com.apple.CoreSimulator.SimDeviceType.Apple-TV-4K-4K)
Apple TV 4K (at 1080p) (com.apple.CoreSimulator.SimDeviceType.Apple-TV-4K-1080p)
Apple Watch - 38mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-38mm)
Apple Watch - 42mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-42mm)
Apple Watch Series 2 - 38mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-Series-2-38mm)
Apple Watch Series 2 - 42mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-Series-2-42mm)
Apple Watch Series 3 - 38mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-Series-3-38mm)
Apple Watch Series 3 - 42mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-Series-3-42mm)
Apple Watch Series 4 - 40mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-Series-4-40mm)
Apple Watch Series 4 - 44mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-Series-4-44mm)
Apple Watch Series 5 - 40mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-Series-5-40mm)
Apple Watch Series 5 - 44mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-Series-5-44mm)

== Runtimes ==
iOS 13.3 (13.3 - 17C45) - com.apple.CoreSimulator.SimRuntime.iOS-13-3
tvOS 13.3 (13.3 - 17K446) - com.apple.CoreSimulator.SimRuntime.tvOS-13-3
watchOS 6.1 (6.1.1 - 17S445) - com.apple.CoreSimulator.SimRuntime.watchOS-6-1

== Devices ==
-- iOS 13.3 --
```

```
iPhone 8 (54589698-0C9F-407D-B21A-83432CABB681) (Shutdown)
iPhone 8 Plus (509B7103-97DB-4AB9-B829-001190ED4B7E) (Shutdown)
iPhone 11 (509BC7C7-9C0E-42FA-8AB2-F5220EBAA13B) (Shutdown)
iPhone 11 Pro (3E8E7E92-66F2-4AF3-A405-23B5FB231DE7) (Shutdown)
iPhone 11 Pro Max (50C15135-1532-44C5-B82C-B327F88F2712) (Shutdown)
iPad Pro (9.7-inch) (B11D5D40-FEA2-4114-B053-E4CFD29D127C) (Shutdown)
iPad (7th generation) (7F8EDE89-74E0-4BAB-B3CA-09E2DAE1F095) (Shutdown)
iPad Pro (11-inch) (04DD3B8A-5B78-48E8-8B22-56796A9CFB73) (Shutdown)
iPad Pro (12.9-inch) (3rd generation) (D811684E-2F3E-4FC6-92EA-39301451F7E5) (Shutdown)

iPad Air (3rd generation) (BBC48526-3922-4C97-BA14-B1888385243A) (Shutdown)
-- tvOS 13.3 --
Apple TV (6680F059-4DE1-430C-B696-228AC27CAA88) (Shutdown)
Apple TV 4K (048E58E8-6A27-4D81-BDEB-8812C610B756) (Shutdown)
Apple TV 4K (at 1080p) (384D5E60-B6B1-481E-BDC3-B7FF8F773412) (Shutdown)
-- watchOS 6.1 --
Apple Watch Series 4 - 40mm (1B98415B-3FDE-401B-A80C-A3551DB207D7) (Shutdown)
Apple Watch Series 4 - 44mm (661838E9-B0BE-42B4-B55E-9A34263B1AEA) (Shutdown)
Apple Watch Series 5 - 40mm (F76D77EF-0932-4164-94BB-9FC757420911) (Shutdown)
Apple Watch Series 5 - 44mm (D86F0BD5-4D38-4537-9C8C-2F5C74E404CA) (Shutdown)
-- Device Pairs --
56795D8F-84E0-4F5A-BA60-517EF25593FF (active, disconnected)
    Watch: Apple Watch Series 5 - 40mm (F76D77EF-0932-4164-94BB-9FC757420911) (Shutdown)
    Phone: iPhone 11 Pro (3E8E7E92-66F2-4AF3-A405-23B5FB231DE7) (Shutdown)
4DDF7790-928A-4D86-B2BC-213F785F5188 (active, disconnected)
    Watch: Apple Watch Series 5 - 44mm (D86F0BD5-4D38-4537-9C8C-2F5C74E404CA) (Shutdown)
    Phone: iPhone 11 Pro Max (50C15135-1532-44C5-B82C-B327F88F2712) (Shutdown)
```

## ideviceinstaller

在Mac中，想要查看（已通过USB连接上的）iOS设备中已安装的应用信息：

- app名称
- app包名= `bundle id`
- app版本
- 等

可以用：

- `ideviceinstaller`
  - 作用：列出已安装的app信息

## 安装ideviceinstaller

```
brew install --HEAD ideviceinstaller
```

安装后就有了：

- `ideviceinstaller`
  - 内部会自动安装额外的依赖
    - `libusb`
    - `libusbmuxd`
    - `libimobiledevice`
      - 其包含多个工具：
        - `idevice_id`
        - 等
      - `libplist`
      - `libtasn1`
      - `libzip`

## 使用

语法：

```
ideviceinstaller -l
```

举例：

```
ideviceinstaller -l
Total: 37 apps
com.suiyi.foodshop1 - 食行生鲜 4911
```

```
com.cisco.anyconnect - AnyConnect 4.6.03052
com.smartisan.reader - 锤子阅读 1311
com.baidu.BaiduMobile - 百度 10.5.5.10
com.ishuyin.iShuYin - 爱书音 1.22
com.evernote.iPhone.Evernote - 印象笔记 358974
com.alipay.iphoneclient - 支付宝 10.1.2.091512
com.autonavi.amap - 高德地图 8.3.0.2104
ctrip.com - 携程旅行 8.3.0
com.Qting.QTTour - 蜻蜓FM 8.0.1.4
com.iflytek.iflyinput - 讯飞输入法 7.0.1815.9602
com.360buy.jdmobile - 京东 7.3.6
com.taobao.tmall - 手机天猫 10948419
com.crifan.voicerecorddemo - 飞语录音Demo 1
org.reactjs.native.example.AwesomeProject - AwesomeProject 1
com.yingwen.xqlv - 中国象棋 1.01.1
com.crifan.WebDriverAgentRunner.xctrunner - WebDriverAgentRunner-Runner 1
com.tencent.xin - 微信 6.7.4.44
com.cnvcs.xiangqi - 中国象棋 1.5.0
com.netease.cloudmusic - 网易云音乐 876
com.tencent.mqq - QQ 7.2.9.404
...
...
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:23:59

# 端口转发

用 iproxy 或 mobileddevice

- 如果想要用 `localhost` (或 `127.0.0.1`)去访问 (已连接到Mac上的iOS设备)
  - 需要做端口转发
    - `iproxy`
      - 安装: `brew install --HEAD libimobiledevice`
      - 语法: `iproxy <local port> <remote port> [udid]`
      - 用法举例:
        - `iproxy 8100 8100`
          - 当前只连接一个iOS设备时, 可以不指定, 忽略 UDID
        - `iproxy 8100 8100 ed94089f3e34d5538065a695bfd03dfbb3c5579`
          - 指定对应设备的UDID
            - `ed94089f3e34d5538065a695bfd03dfbb3c5579` 是此处的iPhone的UDID
          - 可以通过 `idevice_id` 得到
            - `CUR_UDID=$(idevice_id -l | head -n1)`
      - `mobileddevice`
        - 安装: `brew install mobileddevice`
        - 用法:
          - `mobileddevice tunnel 8100 8100`
          - `mobileddevice tunnel -u ed94089f3e34d5538065a695bfd03dfbb3c5579 8100 8100`
            - 同上, 可通过 `-u ios_device_udid`, 指定对应iOS设备

## iproxy

安装:

```
brew install usbmuxd
```

语法

```
~ iproxy --help
usage: iproxy LOCAL_TCP_PORT DEVICE_TCP_PORT [UDID]
```

## mobileddevice

安装:

```
brew install mobileddevice
```

语法：

```

~ mobiledevice help
mobiledevice help
Display this help screen

mobiledevice version [options]
Display program version.
Options:
-r: Include revision identifier

mobiledevice list_devices [options]
Display UDID of each connected devices.
Options:
-t <timeout> : Timeout (in ms) to wait for devices (default: 1)
-n <count> : Limit the number of devices to be printed

mobiledevice list_device_props [options]
List all property names of device.
Options:
-u <udid> : Filter by device UDID (default: first detected device)
-t <timeout> : Timeout (in ms) to wait for devices (default: 1)

mobiledevice get_device_prop [options] <prop_name>
Display value of device property with given name.
Options:
-u <udid> : Filter by device UDID (default: first detected device)
-t <timeout> : Timeout (in ms) to wait for devices (default: 1)

mobiledevice list_apps [options]
Lists all apps installed on device
Options:
-u <udid> : Filter by device UDID (default: first detected device)
-t <timeout> : Timeout (in ms) to wait for devices (default: 1)

mobiledevice list_app_props [options] <bundle_id>
List all property names of app with given bundle id.
Options:
-u <udid> : Filter by device UDID (default: first detected device)
-t <timeout> : Timeout (in ms) to wait for devices (default: 1)

mobiledevice get_app_prop [options] <bundle_id> <prop_name>
Display value of app property with given name.
Options:
-u <udid> : Filter by device UDID (default: first detected device)
-t <timeout> : Timeout (in ms) to wait for devices (default: 1)

```

```
mobiledevice install_app [options] <path_to_app>
  Install app (.app folder) to device
  Options:
    -u <udid> : Filter by device UDID (default: first detected device)
    -t <timeout> : Timeout (in ms) to wait for devices (default: 1)

mobiledevice uninstall_app [options] <bundle_id>
  Uninstall app with given bundle id from device
  Options:
    -u <udid> : Filter by device UDID (default: first detected device)
    -t <timeout> : Timeout (in ms) to wait for devices (default: 1)

mobiledevice tunnel [options] <from_port> <to_port>
  Forward TCP connections to connected device
  Options:
    -u <udid> : Filter by device UDID (default: first detected device)
    -t <timeout> : Timeout (in ms) to wait for devices (default: 1)

mobiledevice get_bundle_id <path_to_app>
  Display bundle identifier of app (.app folder)
```

## idevice\_id

可以用 `idevice_id` 列出当前（Mac中已）连接的（iOS）的 设备ID = UDID

## 安装

```
brew install libimobiledevice
```

安装 `libimobiledevice` 后，根据[GitHub官网](#)介绍，其会包含很多工具：

Utility	Description
<code>idevice_id</code>	List attached devices or print device name of given device
<code>idevicebackup</code>	Create or restore backup for devices (legacy)
<code>idevicebackup2</code>	Create or restore backups for devices running iOS 4 or later
<code>idevicecrashreport</code>	Retrieve crash reports from a device
<code>idevicedate</code>	Display the current date or set it on a device
<code>idevicedebug</code>	Interact with the debugserver service of a device
<code>idevicedebugserverproxy</code>	Proxy a debugserver connection from a device for remote debugging
<code>idevicediagnostics</code>	Interact with the diagnostics interface of a device
<code>ideviceenterrecovery</code>	Make a device enter recovery mode
<code>ideviceimagemounter</code>	Mount disk images on the device
<code>ideviceinfo</code>	Show information about a connected device
<code>idevicename</code>	Display or set the device name
<code>idevicenotificationproxy</code>	Post or observe notifications on a device
<code>idevicepair</code>	Manage host pairings with devices and usbmuxd
<code>ideviceprovision</code>	Manage provisioning profiles on a device
<code>idevicescreenshot</code>	Gets a screenshot from the connected device
<code>idevicesetlocation</code>	Simulate location on device
<code>idevicesyslog</code>	Relay syslog of a connected device

## 使用

举例：

```
> idevice_id -l
```

```
ed94089f3e34d5538065a695bfd03dfbb3c5579
```

如果有多个设备，想要获取第一个，则可以借助 head：

```
> idevice_id -l | head -n1  
ed94089f3e34d5538065a695bfd03dfbb3c5579
```

## 帮助和语法

```
~ idevice_id --help  
Usage: idevice_id [OPTIONS] [UDID]  
Prints device name or a list of attached devices.  
  
If UDID is given, the name of the connected device with that UDID will be retrieved.  
  
-l, --list      list UDIDs of all devices attached via USB  
-n, --network   list UDIDs of all devices available via network  
-d, --debug     enable communication debugging  
-h, --help      prints usage information  
  
Homepage: http://libimobiledevice.org
```

## system\_profiler

system\_profiler: 获取系统信息

举例：

从输出中解析出：

### 真机设备ID

```
system_profiler SPUSBDataType
```

注：

此处输出的是很多非常多的信息：

```
~ system_profiler SPUSBDataType
2020-04-29 14:50:16.086 system_profiler[46290:995081] SPUSBDevice: IOCreatePlugInInterface
ForService failed 0xe00002be
2020-04-29 14:50:16.086 system_profiler[46290:995081] SPUSBDevice: IOCreatePlugInInterface
ForService failed 0xe00002be
2020-04-29 14:50:16.087 system_profiler[46290:995081] SPUSBDevice: IOCreatePlugInInterface
ForService failed 0xe00002be
2020-04-29 14:50:16.087 system_profiler[46290:995081] SPUSBDevice: IOCreatePlugInInterface
ForService failed 0xe00002be
2020-04-29 14:50:16.088 system_profiler[46290:995081] SPUSBDevice: IOCreatePlugInInterface
ForService failed 0xe00002be
2020-04-29 14:50:16.089 system_profiler[46290:995081] SPUSBDevice: IOCreatePlugInInterface
ForService failed 0xe00002be
USB:
USB 3.1 Bus:

Host Controller Driver: AppleIntelCNLUSBXHCI
PCI Device ID: 0x9ded
PCI Revision ID: 0x0030
PCI Vendor ID: 0x8086

USB2.1 Hub:

Product ID: 0x0610
Vendor ID: 0x05e3 (Genesys Logic, Inc.)
Version: 6.53
Speed: Up to 480 Mb/sec
Manufacturer: GenesysLogic
Location ID: 0x14200000 / 1
Current Available (mA): 500
Current Required (mA): 100
Extra Operating Current (mA): 0
```

**iPhone:**

```
Product ID: 0x12a8
Vendor ID: 0x05ac (Apple Inc.)
Version: 7.02
Serial Number: ed94089f3e34d5538065a695bfd03dfbb3c5579
Speed: Up to 480 Mb/sec
Manufacturer: Apple Inc.
Location ID: 0x14230000 / 16
Current Available (mA): 500
Current Required (mA): 500
Extra Operating Current (mA): 0
Sleep current (mA): 500
```

**USB Composite Device:**

```
Product ID: 0x0002
Vendor ID: 0x0603 (Novatek Microelectronics Corp.)
Version: 16.12
Speed: Up to 1.5 Mb/sec
Manufacturer: SINO WEALTH
Location ID: 0x14220000 / 8
Current Available (mA): 500
Current Required (mA): 100
Extra Operating Current (mA): 0
```

**USB 3.1 Bus:**

```
Host Controller Driver: AppleUSBXHCITR
PCI Device ID: 0x15ec
PCI Revision ID: 0x0006
PCI Vendor ID: 0x8086
Bus Number: 0x00
```

**USB 3.1 Bus:**

```
Host Controller Driver: AppleUSBXHCITR
PCI Device ID: 0x15ec
PCI Revision ID: 0x0006
PCI Vendor ID: 0x8086
Bus Number: 0x01
```

**USB3.1 Hub:**

```
Product ID: 0x0626
Vendor ID: 0x05e3 (Genesys Logic, Inc.)
Version: 6.53
Speed: Up to 5 Gb/sec
Manufacturer: GenesysLogic
Location ID: 0x01100000 / 1
Current Available (mA): 900
Current Required (mA): 0
```

```
Extra Operating Current (mA): 0
```

iBridge Bus:

```
Host Controller Driver: AppleUSBVHCIBCE
```

Touch Bar Backlight:

```
Product ID: 0x8102
Vendor ID: 0x05ac (Apple Inc.)
Version: 2.01
Serial Number: 0000000000000000
Manufacturer: Apple Inc.
Location ID: 0x80700000
```

Touch Bar Display:

```
Product ID: 0x8302
Vendor ID: 0x05ac (Apple Inc.)
Version: 2.01
Serial Number: 0000000000000000
Manufacturer: Apple Inc.
Location ID: 0x80600000
```

Apple Internal Keyboard / Trackpad:

```
Product ID: 0x027b
Vendor ID: 0x05ac (Apple Inc.)
Version: 9.27
Serial Number: FM7845603R8J3VXAH+TVZ
Speed: Up to 480 Mb/sec
Manufacturer: Apple Inc.
Location ID: 0x80500000 / 8
Current Available (mA): 500
Current Required (mA): 500
Extra Operating Current (mA): 0
Built-In: Yes
```

Headset:

```
Product ID: 0x8103
Vendor ID: 0x05ac (Apple Inc.)
Version: 2.04
Serial Number: 000000000000
Manufacturer: Apple
Location ID: 0x80400000
```

Ambient Light Sensor:

```
Product ID: 0x8262
Vendor ID: 0x05ac (Apple Inc.)
Version: 2.01
Serial Number: 000000000000
Manufacturer: Apple Inc.
Location ID: 0x80300000
```

**FaceTime HD Camera (Built-in):**

```
Product ID: 0x8514
Vendor ID: 0x05ac (Apple Inc.)
Version: 2.01
Serial Number: CC28493XQ52J3Y324
Manufacturer: Apple Inc.
Location ID: 0x80200000
```

**Apple T2 Controller:**

```
Product ID: 0x8233
Vendor ID: 0x05ac (Apple Inc.)
Version: 2.01
Serial Number: 0000000000000000
Manufacturer: Apple Inc.
Location ID: 0x80100000
```

## 屏幕（宽度和高度等）信息

```
system_profiler SPDDisplaysDataType
Graphics/Displays:

Intel Iris Plus Graphics 655:

Chipset Model: Intel Iris Plus Graphics 655
Type: GPU
Bus: Built-In
VRAM (Dynamic, Max): 1536 MB
Vendor: Intel
Device ID: 0x3ea5
Revision ID: 0x0001
Metal: Supported, feature set macOS GPUFamily2 v1
Displays:
Color LCD:
Display Type: Built-In Retina LCD
Resolution: 2560 x 1600 Retina
Framebuffer Depth: 24-Bit Color (ARGB8888)
Main Display: Yes
Mirror: Off
```

```
Online: Yes
Rotation: Supported
Automatically Adjust Brightness: No
```

详见：

【已解决】Mac中获取iPhone的分辨率宽高等屏幕信息

## 获取Mac的序列号

```
system_profiler SPHardwareDataType | grep Serial
Serial Number (system): C02Y3N10JHC8
```

详见：

【已解决】Mac中如何获取笔记本的序列号

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:23:59

# instruments

可以用 instruments 列出当前所有苹果的设备

举例：

```
~ instruments -s
CoreData: annotation: Failed to load optimized model at path '/Applications/Xcode.app/Contents/Applications/Instruments.app/Contents/Frameworks/InstrumentsPackaging.framework/Versions/A/Resources/XRPackageModel.momd/XRPackageModel 9.0.omo'
Known Devices:
limao的MacBook Pro [F9089371-1060-5CE3-99BB-81741693BE80]
Crifan iPhone6 (12.4.5) [ed94089f3e34d5538065a695bfd03dfbb3c5579]
Apple TV (13.3) [6680F059-4DE1-430C-B696-228AC27CAA88] (Simulator)
Apple TV 4K (13.3) [048E58E8-6A27-4D81-BDEB-8812C610B756] (Simulator)
Apple TV 4K (at 1080p) (13.3) [384D5E60-B6B1-481E-BDC3-B7FF8F773412] (Simulator)
Apple Watch Series 4 - 40mm (6.1.1) [1B98415B-3FDE-401B-A80C-A3551DB207D7] (Simulator)
Apple Watch Series 4 - 44mm (6.1.1) [661838E9-B0BE-42B4-B55E-9A34263B1AEA] (Simulator)
iPad (7th generation) (13.3) [7F8EDE89-74E0-4BAB-B3CA-09E2DAE1F095] (Simulator)
iPad Air (3rd generation) (13.3) [BBC48526-3922-4C97-BA14-B1888385243A] (Simulator)
iPad Pro (11-inch) (13.3) [04DD3B8A-5B78-48E8-8B22-56796A9CFB73] (Simulator)
iPad Pro (12.9-inch) (3rd generation) (13.3) [D811684E-2F3E-4FC6-92EA-39301451F7E5] (Simulator)
iPad Pro (9.7-inch) (13.3) [B11D5D40-FEA2-4114-B053-E4CFD29D127C] (Simulator)
iPhone 11 (13.3) [509BC7C7-9C0E-42FA-8AB2-F5220EBAA13B] (Simulator)
iPhone 11 Pro (13.3) [3E8E7E92-66F2-4AF3-A405-23B5FB231DE7] (Simulator)
iPhone 11 Pro (13.3) + Apple Watch Series 5 - 40mm (6.1.1) [F76D77EF-0932-4164-94BB-9FC757420911] (Simulator)
iPhone 11 Pro Max (13.3) [50C15135-1532-44C5-B82C-B327F88F2712] (Simulator)
iPhone 11 Pro Max (13.3) + Apple Watch Series 5 - 44mm (6.1.1) [D86F0BD5-4D38-4537-9C8C-2F5C74E404CA] (Simulator)
iPhone 8 (13.3) [54589698-0C9F-407D-B21A-83432CABB681] (Simulator)
iPhone 8 Plus (13.3) [509B7103-97DB-4AB9-B829-001190ED4B7E] (Simulator)
Known Templates:
"Activity Monitor"
"Allocations"
"App Launch"
"Blank"
"Core Animation"
"Core Data"
"Counters"
"Energy Log"
"File Activity"
"Game Performance"
"Leaks"
"Metal System Trace"
"Network"
"SceneKit"
"SwiftUI"
"System Trace"
"Time Profiler"
```

"**Zombies**"

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:23:59

## security

举例：

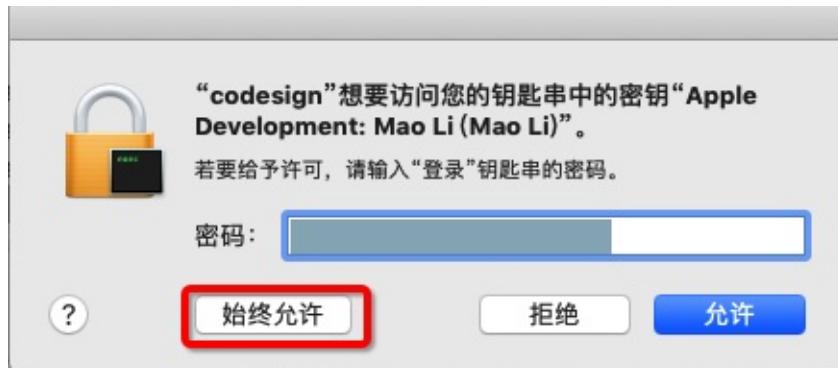
```
security unlock-keychain -p xxx ~/Library/Keychains/login.keychain-db
```

暂时无效，有待后续深入研究

之前用：

```
# 解锁keychain, 以便可以正常的签名应用,  
PASSWORD="replace-with-your-password"  
security unlock-keychain -p $PASSWORD ~/Library/Keychains/login.keychain-db  
# 获取设备的UDID  
CUR_UDID=$(idevice_id -l | head -n1)  
# 运行测试  
xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination "id=$  
$CUR_UDID" test
```

实现自动化测试，结果却还会弹框让输入密码：



-» 说明之前的： `security unlock-keychain` 并没有起效果

-» 具体原因，有待深究。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:23:59

## 移动端

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:23:59

# iOS

## iOS自动化

详见完整教程：

[移动端自动化测试概览](#)

中的

[iOS自动化测试利器：facebook-wda](#)

## XCTest

- iOS最新测试框架是：`XCTest`
  - 别称：`XCUITest`

下面列出一些常用的部分：

- 用户界面
  - [User Interface Tests | Apple Developer Documentation](#)
    - 其中常用的部分是：
      - `XCUIScreen`
        - A physical screen attached to a device
      - `XCUIScreenshot`
        - A captured image of a screen, app, or UI element state.
      - `XCUIDevice`
        - Simulates physical buttons, device orientation, and Siri interaction for an iOS device.
      - `XCUISiriService`
        - Simulates a device's Siri interface.
      - `XCUIRemote`
        - Simulates interaction with a physical remote control.

关于具体细节如下：

- 设备
  - [XCUIDevice - XCTest | Apple Developer Documentation](#)
    - `sharedDevice`
      - The current device.
- 远程
  - [XCUIRemote - XCTest | Apple Developer Documentation](#)
    - `pressButton :`
      - Sends a momentary press of a button on a physical remote control.

## 一些心得

找接口和函数时，可以充分利用官网自带的搜索

比如想要找哪些类中有mainScreen，可以搜：

[mainScreen](#)

<https://developer.apple.com/search/?q=mainScreen>

The screenshot shows the Apple Developer Documentation search interface. The search bar at the top contains the query 'mainScreen'. Below the search bar, there are tabs for 'All', 'Reference', 'Guides', 'Sample Code', and 'Videos', with 'All' being selected. The results section displays 7 results for 'mainScreen'. The first result is highlighted with a red box and is for the UIKit class. It shows the method signature 'UIKit > UIScreen > mainScreen', a brief description 'Returns the screen object representing the device's screen.', and language support for iOS, macOS, Catalyst, tvOS, and Objective-C. The second result is also highlighted with a red box and is for the AppKit class. It shows the method signature 'AppKit > NSScreen > mainScreen', a brief description 'Returns the screen object containing the window with the keyboard focus.', and language support for macOS and Objective-C. The third result is for the XCTest class.

可以看到我们希望找的有2处：

- `mainScreen`
  - `UIKit -> UIScreen -> mainScreen`
    - Returns the screen object representing the device's screen.
- `mainScreen`
  - `XCTest -> XCUIScreen -> mainScreen`
    - The current device's main screen.

官网文档分语言的 -》 Swift和Objective-C 接口略有不同的

- `XCUIScreen`
  - Swift
    - `XCUIScreen - XCTest | Apple Developer Documentation`

- <https://developer.apple.com/documentation/xctest/xcuiscreen>
- Objective-C
  - XCUIScreen - XCTest | Apple Developer Documentation
  - <https://developer.apple.com/documentation/xctest/xcuiscreen?language=objc>

不过，总体上内容是一致的：

- mainScreen
  - The current device's main screen.
- screens
  - The current device's active screens.

只不过Swift和OC的写法不太一样而已。

注：之前见过个别函数和接口好像功能上略有不同。

目前没找到。等找到了。再补充。

总之：注意看文档时所选择的语言，是Swift还是ObjC，不要搞错就好。

## 测试Screen相关内容：XCUIScreen和UIScreen

- XCUIScreen
  - 文档
    - XCUIScreen - XCTest | Apple Developer Documentation
    - <https://developer.apple.com/documentation/xctest/xcuiscreen?language=objc>
- UIScreen
  - 文档
    - UIScreen - UIKit | Apple Developer Documentation
    - <https://developer.apple.com/documentation/uikit/uiscreen?language=objc>

之所以要注意此处有2个Screen的原因是：

之前想要找，除了scale之外的bounds属性，最后发现：XCUIScreen是没有的

所以代码：

```
return [XCUIScreen mainScreen nativeScale];
return [XCUIScreen mainScreen bounds];
return [XCUIScreen mainScreen nativeBounds];
```

会报错。要改为：

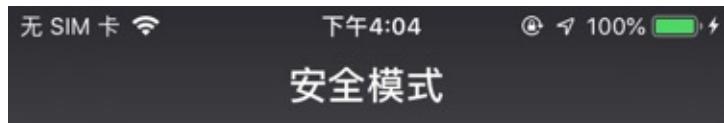
```
return [UIScreen mainScreen nativeScale];
return [UIScreen mainScreen bounds];
return [UIScreen mainScreen nativeBounds];
```

才可以。

## 微信

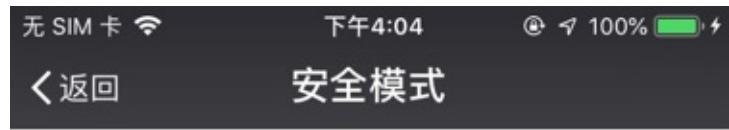
连续多次崩溃会进入安全模式

iOS中微信如果检测到连续崩溃了多次后，会进入安全模式：



当前检测出微信连续异常，你可以尝试以下方法修复：

下一步



< 返回

## 安全模式



清理缓存会清理你的手机本地缓存文件，但不会清理你的消息数据，使用后需要重新登录微信

清理缓存

不清理缓存，进入下一步

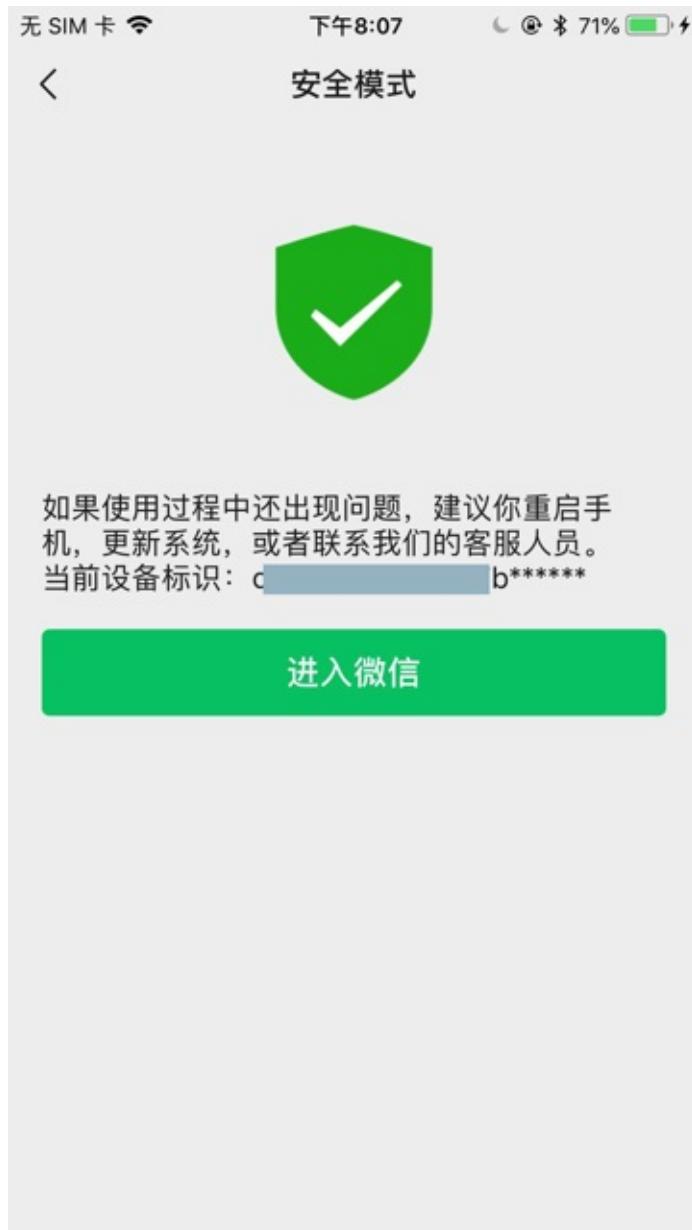
下一步





然后才会恢复正常微信界面。

后来，换了iPhone 6P，多次调试期间，虽然微信没怎么崩溃，但是也遇到类似的问题：



如果要用代码自动化操作实现上述步骤，则可以参考：

[安全模式 · iOS 自动化测试利器：facebook-wda](#)

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook 最后更新：2020-08-09 10:23:59

# iPhone

iOS 设备中普通用户接触最多的就是 iPhone 了。

此处整理相关开发心得。

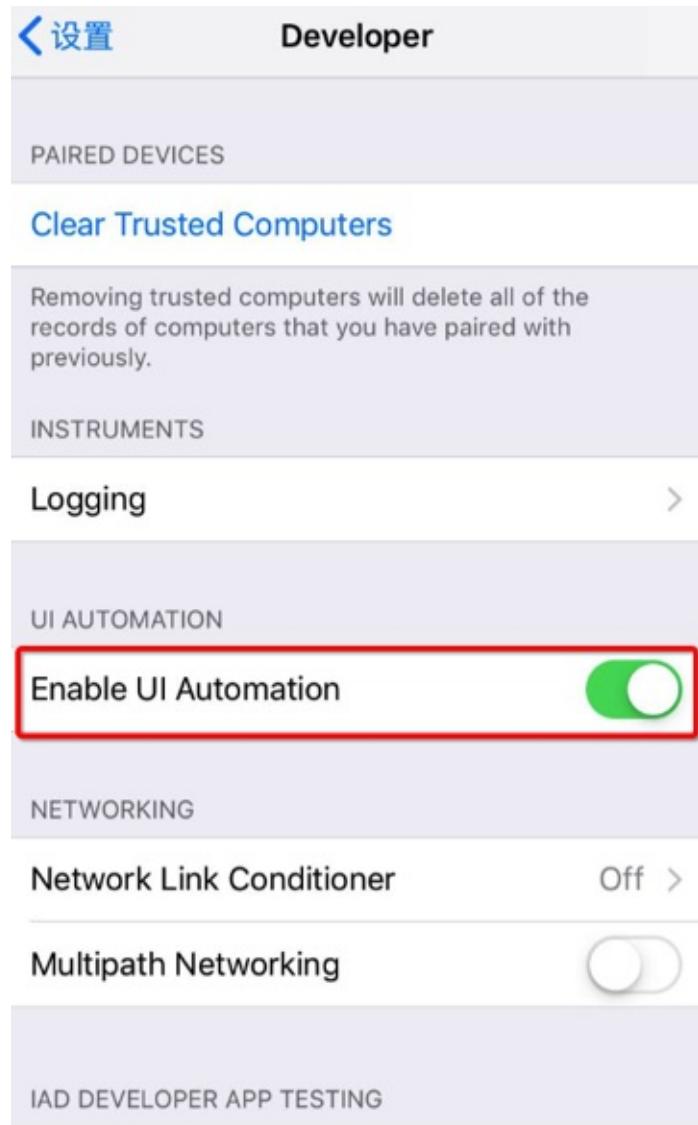
## 开发相关设置

新版iPhone中的设置中有个 开发者，有很多开发相关的设置。

其中和自动化测试有关的是：

设置 -> 开发者 -> Enable UI Automation





对于后续自动化测试，或许有用。

## iPhone真机的log日志查看

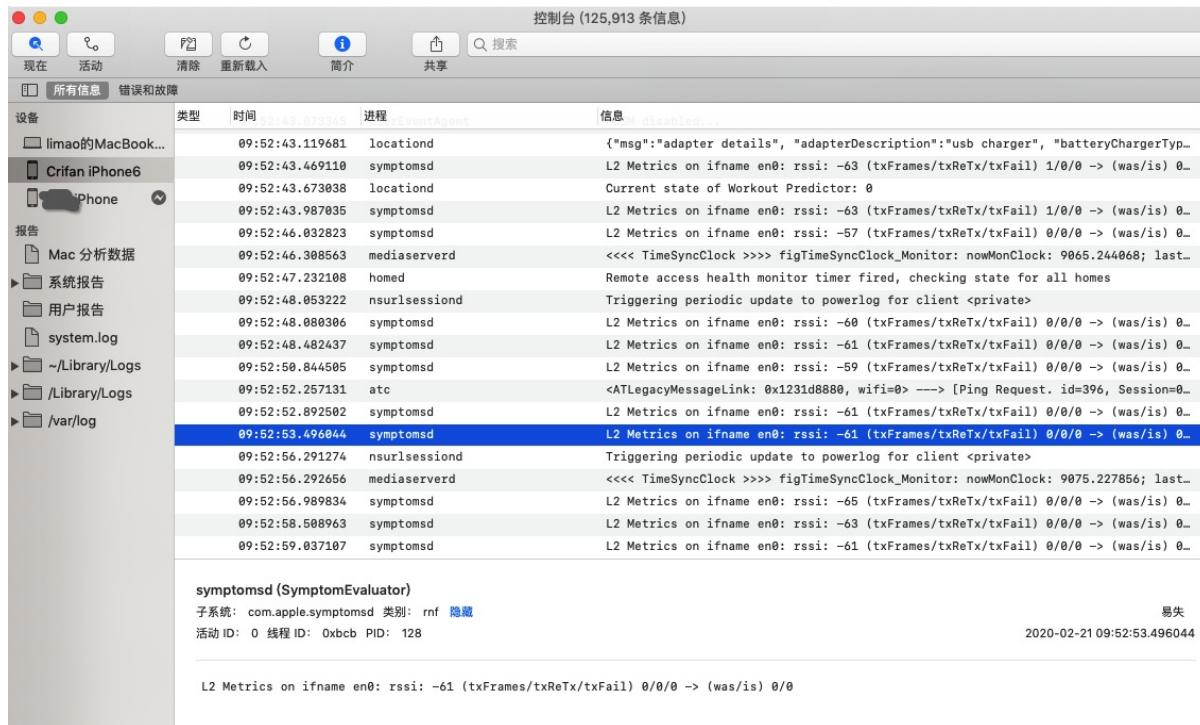
可以通过 控制台 查看iPhone真机的Log日志

如何打开控制台：

- 启动台Launch Pad -> 其他 -> 控制台

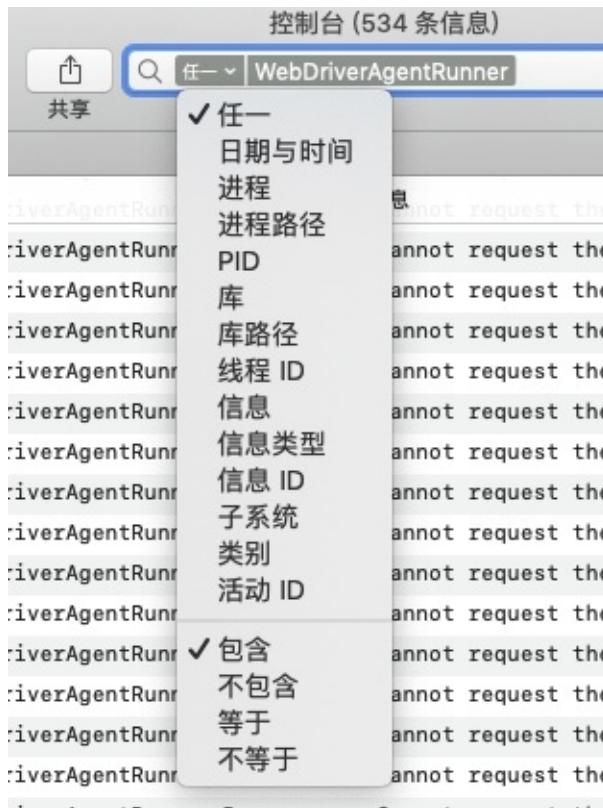
- - XCode->Window->Devices and Simulators->Devices

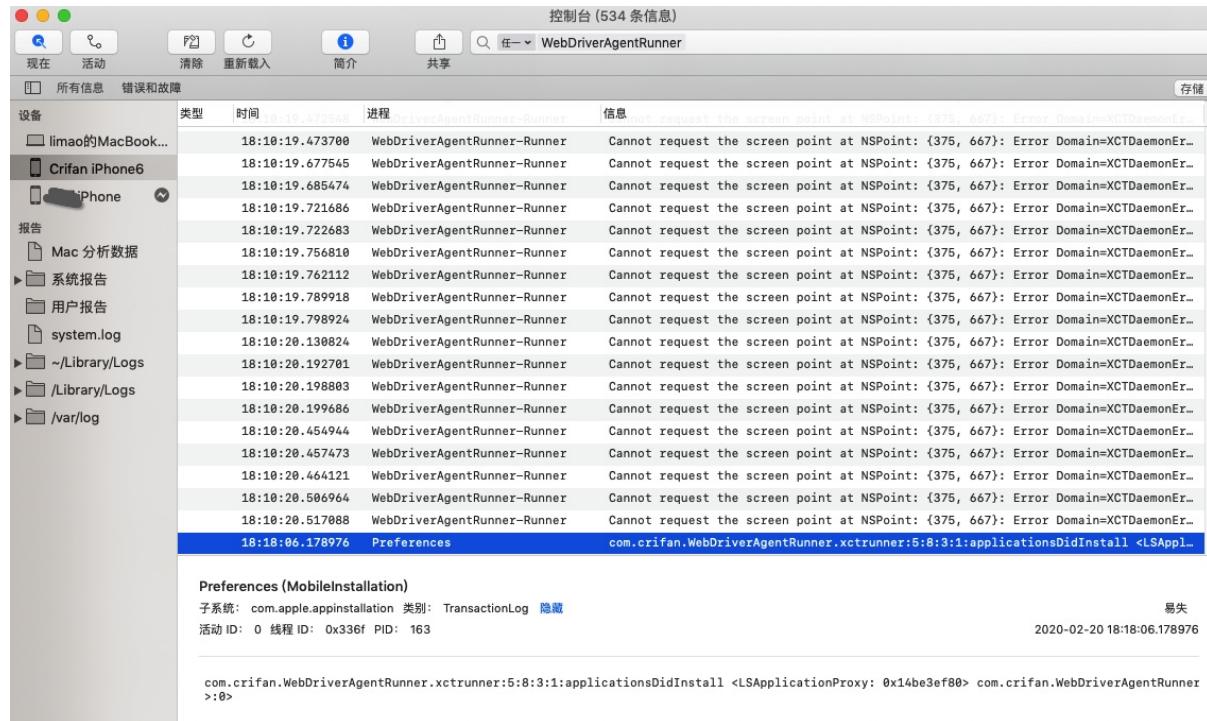
启动后，即可看到iPhone真机的log日志了：



也支持条件过滤，比如：

任一包含： WebDriverAgentRunner





## 关闭悬浮球

iOS自动化测试期间，记得要关闭：悬浮球

【已解决】iPhone中关闭全屏显示的悬浮球

否则有时候会误触发，影响自动化测试

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:23:59

## 附录

下面列出相关参考资料。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:23:59

# 文档

此处整理出有用的苹果相关开发文档，供需要时查阅。

## iOS 旧文档

- 举例
  - [Displays - iOS Device Compatibility Reference](#)

## iOS 新文档

统一后的 = 新的

- 举例
  - [UIScreen - UIKit | Apple Developer Documentation](#)
- 且还分2种语言
  - Swift
    - <https://developer.apple.com/documentation/uikit/uiscreen>
    - 截图

The screenshot shows the Apple Developer Documentation for the `UIScreen` class. At the top, there's a navigation bar with links like 'Documentation', 'UIKit', 'Windows and Screens', and 'UIScreen'. Below the navigation bar, the 'Language' dropdown is highlighted with a red arrow, showing 'Swift' as the selected option. The main content area starts with the class name `UIScreen` and a brief description: 'An object that defines the properties associated with a hardware-based display.' It then lists the 'Declaration' code: `class UIScreen : NSObject`. In the bottom right corner, there's a sidebar titled 'On This Page' with links to 'Declaration', 'Overview', 'Topics', 'Relationships', and 'See Also'.

- Objective-C
  - <https://developer.apple.com/documentation/uikit/uiscreen?language=objc>
  - 截图

**UIScreen**

An object that defines the properties associated with a hardware-based display.

**Declaration**

```
@interface UIScreen : NSObject
```

**Overview**

iOS devices have a main screen and zero or more attached screens. A tvOS device has a main screen for the television connected to the device. Use this class to obtain screen objects for each display attached to the device. Each screen object defines the bounds rectangle for the associated display and other interesting properties such as its brightness. In iOS 8 and later, a

■

- 说明
  - 其中内容大体类似，主要是语法不同
    - 不过有时候细节也不太一样
- 举例
  - Objective-C 中 `UIDeviceBatteryState` 中，找不到枚举值的常量值定义

### [UIDeviceBatteryState - UIKit | Apple Developer Documentation](#)

```
typedef enum UIDeviceBatteryState : NSInteger {
    ...
} UIDeviceBatteryState;
```

- Constants
  - `UIDeviceBatteryStateUnknown`
    - The battery state for the device cannot be determined.
  - `UIDeviceBatteryStateUnplugged`
    - The device is not plugged into power; the battery is discharging.
  - `UIDeviceBatteryStateCharging`
    - The device is plugged into power and the battery is less than 100% charged.
  - `UIDeviceBatteryStateFull`
    - The device is plugged into power and the battery is 100% charged.

-》想要知道对应的枚举值定义的int值

-》

### [UIDeviceBatteryStateUnknown - UIDeviceBatteryState | Apple Developer Documentation](#)

The battery state for the device cannot be determined.

`UIDeviceBatteryStateUnknown`

-》没看到定义的值

-》无意间发现，换Swift语言后：

[UIDevice.BatteryState - UIDevice | Apple Developer Documentation](#)

-》点击具体的某个定义，都可以看到具体常量枚举值定义：

- case unknown
  - The battery state for the device cannot be determined.
    - [UIDevice.BatteryState.unknown - UIDevice.BatteryState | Apple Developer Documentation](#)
    - <https://developer.apple.com/documentation/uikit/uidevice/batterystate/unknown>
    - case unknown = 0
- case unplugged
  - The device is not plugged into power; the battery is discharging.
    - [UIDevice.BatteryState.unplugged - UIDevice.BatteryState | Apple Developer Documentation](#)
    - <https://developer.apple.com/documentation/uikit/uidevice/batterystate/unplugged>
    - case unplugged = 1
- case charging
  - The device is plugged into power and the battery is less than 100% charged.
    - [UIDevice.BatteryState.charging - UIDevice.BatteryState | Apple Developer Documentation](#)
    - <https://developer.apple.com/documentation/uikit/uidevice/batterystate/charging>
    - case charging = 2
- case full
  - The device is plugged into power and the battery is 100% charged.
    - [UIDevice.BatteryState.full - UIDevice.BatteryState | Apple Developer Documentation](#)
    - <https://developer.apple.com/documentation/uikit/uidevice/batterystate/full>
    - case full = 3

-》总结来说：

此处是 ObjC 中看不到 `UIDeviceBatteryState` 的细节的枚举值的定义

而换成 Swift 的 `UIDevice.BatteryState`，就可以看到具体的枚举的定义的常量的值了

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:23:59

## 参考资料

- 【已解决】Mac中获取iPhone的分辨率宽高等屏幕信息
- 【已解决】Mac中如何获取笔记本的序列号
- 【已解决】扩展Python的facebook-wda源码以返回更新屏幕相关信息
- 【已解决】Python中wda代码报错： Invalid type in JSON write NSConcreteValue
- 【已解决】iOS中ObjC中如何使用CGRect类型的变量
- 【已解决】XCode中wda代码报错： No visible interface for XCUIScreen declares the selector bounds
- 【已解决】Mac中用brew安装最新的libimobiledevice
- 【已解决】Mac中brew install usbmuxd期间./autogen.sh出错： ./configure syntax error near unexpected token libplist
- 【已解决】Mac中找不到idevice\_id即idevice\_id not found
- 【无需解决】Mac中iproxy端口转发连接iPhone6真机失败： Error connecting to device!
- 【未解决】Mac中用facebook-wda操作iOS真机iPhone6
- 【已解决】iPhone中关闭全屏显示的悬浮球
- 
- 移动端自动化测试概览
- iOS自动化测试利器：facebook-wda
- 
- 使用自定义 WDA 服务器 - Appium
- 使用 Python 库 facebook-wda 完成网易云音乐 iOS 客户端的自动化测试 (示例) · TesterHome
- Displays - iOS Device Compatibility Reference
- UIScreen - UIKit | Apple Developer Documentation
- UIDeviceBatteryState - UIKit | Apple Developer Documentation
- UIDeviceBatteryStateUnknown - UIDeviceBatteryState | Apple Developer Documentation
- UIDevice.BatteryState - UIDevice | Apple Developer Documentation
- iOS真机安装WebDriverAgent | Vicの博客
- libimobiledevice/libimobiledevice: A cross-platform protocol library to communicate with iOS devices
- idevice\_id command man page - libimobiledevice-utils
- Appium for mac iOS环境配置 - 简书
- ios - How to check device id of iPhone simulator? - Stack Overflow
- Xcode 工具链 - 简书
- 

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:23:59