

目录

前言	1.1
概述	1.2
通用	1.3
IDE	1.3.1
苹果开发者账号	1.3.2
桌面端	1.4
ideviceinstaller	1.4.1
libimobiledevice	1.4.2
idevice_id	1.4.2.1
ideviceinfo	1.4.2.2
端口转发	1.4.3
system_profiler	1.4.4
instruments	1.4.5
security	1.4.6
移动端	1.5
iOS	1.5.1
iPhone	1.5.1.1
子教程	1.6
附录	1.7
文档	1.7.1
参考资料	1.7.2

苹果相关开发总结

- 最新版本: v1.4
- 更新时间: 20221104

简介

总结Apple苹果相关领域的开发经验和心得，包括通用的IDE和开发者账号，桌面端的各种工具，比如ideviceinstaller、libimobiledevice的idevice_id、ideviceinfo, iproxy或mobiledevice的端口转发、以及system_profiler、instruments、security；以及移动端的iOS系统和iPhone设备开发，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；以及相关子教程；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：

HonKit源码

- [crifan/apple_develop_summary](#): 苹果相关开发总结

如何使用此HonKit源码去生成发布为电子书

详见：[crifan/honkit_template](#): demo how to use crifan honkit template and demo

在线浏览

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

版权和用途说明

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

各种技术类教程，仅作为学习和研究使用。请勿用于任何非法用途。如有非法用途，均与本人无关。

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 150+ 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.org, 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：

2022-11-04 15:04:09

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.org, 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:
2022-11-02 22:34:11

通用

此处介绍苹果开发涉及到的通用的内容。

crifan.org, 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

IDE

苹果开发常用的IDE有：

- 苹果自家的：[Xcode](#)
 - 详见独立子教程
 - [Xcode开发心得](#)
- 第三方的：[AppCode](#)
 - 官网
 - [AppCode: Smart Swift/Obj-C IDE for iOS & macOS Development](#)
 - [Smart IDE for iOS/macOS development](#)

crifan.org, 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：
2022-11-03 09:46:35

苹果开发者账号

做苹果的iOS方面的开发，前提是需要有个 苹果开发者账号，一般也被称为 Apple ID = 苹果开发者ID

这个账号是需要购买的，最便宜的个人版，一年也要99美元的。

有了这个苹果开发者账号后，才可以发布app。

而同时，也会享有一些额外的好处。

提前使用新版本

注册了苹果开发者账号后，可以有机会在苹果正式发布Xcode, iOS等最新版本之前，可以去下载试用beta版本

Apple Developer Center特殊待遇：可以先下载一些beta的开发环境

登录

[Apple Developer Center](#)

之后，就可以看到里面有很多在正式发布之前的供开发者内测的beta版的iOS, Xcode等资源：

The screenshot shows the Apple Developer Center's 'Downloads' section. At the top, there's a banner with a warning icon and text: 'Important: You must have the latest beta version of Xcode 8 installed on your Mac before using a Restore Image to install iOS or tvOS beta software on a device.' Below this, a note states: 'Your use of Apple beta software is subject to and licensed only under the terms and conditions of the Apple Developer Program License Agreement, including any applicable consent to collect diagnostic data set forth therein. If you have not agreed to the Apple Developer Program License Agreement, you are not permitted to use this software. Refer to the software installation guides for complete instructions.' A table titled 'Featured Downloads' lists the 'Xcode 8 beta 4' entry, which includes the SDIs for iOS 10 beta, macOS 10.12 beta, watchOS 3 beta, and tvOS 10 beta. The table columns are 'Build', 'Date', and 'Download'. The 'Release Notes' link points to '851880' and the date is 'Aug 1, 2016'.

Build	Date	Download
Release Notes 851880	Aug 1, 2016	Download

Featured Downloads

	Build	Date	
Xcode 8 beta 4 Includes the SDKs for iOS 10 beta, macOS 10.12 beta, watchOS 3 beta, and tvOS 10 beta.	Release Notes 8S188o	Aug 1, 2016	Download
macOS 10.12 beta 4 Refer to the release notes for complete installation instructions.	Release Notes 16A270f	Aug 1, 2016	Download
iOS 10 beta 4 Preferred Install Configuration Profile directly on any iOS device and receive OTA updates.	Release Notes 14A5322e	Aug 1, 2016	Download
iOS Restore Images Install via iTunes.			See all
watchOS 3 beta 4 Install configuration profile directly on your device to receive OTA updates.	Release Notes 14S5302d	Aug 1, 2016	Download
tvOS 10 beta 4 Preferred For Apple TV (4th generation). Apple TV (3rd generation) and earlier are not supported. Install configuration profile directly on your device to receive OTA updates.	Release Notes 14T5308d	Aug 1, 2016	Download
tvOS Restore Image Install via iTunes.			Download
macOS Server 5.2 beta 3	Release Notes 16S1124f	Aug 1, 2016	Download

Featured Downloads

	Build	Date	
tvOS restore image Install via iTunes.			Download
macOS Server 5.2 beta 3	Release Notes 16S1124f	Aug 1, 2016	Download
Apple Configurator 2.3 beta 2 Mac app for deploying iOS devices and Apple TV in your school or business.	Release Notes 3D34	Jul 18, 2016	Download
MobileDevice Installer Required Install the latest MobileDevice installer before using Apple Configurator 2.3 beta.			Download

Beta Software

	Build	Date	
Safari 10 beta 4 Safari 10 beta for OS X El Capitan and OS X Yosemite.	Release Notes 602.143	Aug 1, 2016	See all
Parallax Exporter Use the Parallax Exporter plug-in to preview a layered image while working in Adobe Photoshop.			See all
Parallax Previewer Use Parallax Previewer to preview layered Photoshop files, assemble individual image layers from PNG files and preview the parallax effect, or preview layered images exported by the Parallax Exporter plug-in.	1A68	Jan 25, 2016	Download

crifan.org, 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：
2022-11-02 22:33:10

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.org, 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

ideviceinstaller

TODO:

- 【已解决】 ideviceinstaller卸载iPhone中app后导致无法快捷键截图
- 【已解决】 Mac中如何获取到iPhone中自带应用设置的bundle ID

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
 - 如果单独安装则是：

```
brew install libimobiledevice
```

- 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
...
...
```

获取iPhone中已安装app信息：包名等

- `ideviceinstaller -u {iOSDeviceId} -l -o list_system`
 - 显示系统的app
- `ideviceinstaller -u {iOSDeviceId} -l -o list_user`
 - 显示已安装的用户的app
 - 等价于：
 - `ideviceinstaller -u {iOSDeviceId} -l`
- `ideviceinstaller -u {iOSDeviceId} -l -o list_all`
 - 显示所有（用户+系统）的app

-> 如果只连接了一个iOS设备（比如，只有一个iPhone），则可以省略 -u 参数

```
ideviceinstaller -l -o list_system
ideviceinstaller -l -o list_user
ideviceinstaller -l -o list_all
```

举例：

用户安装的程序：

```
→ ~ ideviceinstaller -u ed94089f3e34d5538065a695bfd03dfbb3c5579 -l
```

```

CFBundleIdentifier, CFBundleVersion, CFBundleDisplayName
com.suiyi.foodshop1, "4911", "食行生鲜"
com.cisco.anyconnect, "4.6.03052", "AnyConnect"
com.baidu.BaiduMobile, "10.5.5.10", "百度"
com.ishuyin.iShuYin, "1.22", "爱书音"
com.evernote.iPhone.Evernote, "358974", "印象笔记"
com.alipay.iphoneclient, "10.1.2.091512", "支付宝"
ctrip.com, "8.3.0", "携程旅行"
com.Qting.QTTour, "8.0.1.4", "蜻蜓FM"
com.360buy.jdmobile, "7.3.6", "京东"
com.taobao.tmall, "10948419", "手机天猫"
com.tencent.xin, "6.7.4.44", "微信"
im.workhub.app, "2016.7.31", "Workhub"
com.netease.cloudmusic, "876", "网易云音乐"
com.tencent.mqq, "7.2.9.404", "QQ"
developer.apple.wwdc-Release, "801.5.2", "Developer"
com.ss.iphone.ugc.Aweme, "114011", "抖音短视频"
com.3WRHBBBW4.com.rileystestut.AltStore, "1", "AltStore"
rn.notes.best, "11122019", "爱思极速版"

```

系统自带（和后来已安装的，被视为系统的）应用 -> 可以查看到系统的应用：

```

→ ~ ideviceinstaller -u ed94089f3e34d5538065a695bfd03dfbb3c5579 -l -o list_system
CFBundleIdentifier, CFBundleVersion, CFBundleDisplayName
com.apple.MobileAddressBook, "1.0", "通讯录"
com.apple.weather, "1.0", "天气"
com.apple.VoiceMemos, "2.0", "语音备忘录"
com.apple.mobilecal, "1.0", "日历"
com.apple.Home, "1", "家庭"
com.apple.reminders, "1.0", "提醒事项"
com.apple.Maps, "7.0", "地图"
com.apple.mobiledmail, "3445.104.12", "邮件"
com.apple.mobilenotes, "1349.30", "备忘录"
com.apple_AXUIViewService, "1", "AXUIViewService"
com.apple.AccountAuthenticationDialog, "1.0", "AccountAuthenticationDialog"
com.apple.AdPlatformsDiagnostics, "1", "Ad Platforms Diagnostics"
com.apple.AppStore, "1", "App Store"
com.apple.AskPermissionUI, "1.0", "AskPermissionUI"
com.apple.CTCarrierSpaceAuth, "1", "CTCarrierSpaceAuth"
com.apple.camera, "1", "相机"
com.apple.CheckerBoard, "1", "CheckerBoard"
com.apple.CompassCalibrationViewService, "1.0", "CompassCalibrationViewService"
com.apple.sidecar.camera, "38", "Continuity Camera"
com.apple.CoreAuthUI, "425.270.3", "用户鉴定"
com.apple.datadetectors.DDAActionsService, "331.18", "DDActionsService"
com.apple.carkit.DNDBuddy, "1", "Do Not Disturb While Driving"
com.apple.DataActivation, "1.0", "DataActivation"
com.apple.DemoApp, "1.0", "DemoApp"
com.apple.Diagnostics, "1", "诊断信息"
com.apple.DiagnosticsService, "1", "DiagnosicsService"
com.apple.family, "1", "Family"
com.apple.appleseed.FeedbackAssistant, "390.11", "Feedback"
com.apple.fieldtest, "1.0", "FieldTest"
com.apple.mobileme.fmip1, "500", "查找 iPhone"
com.apple.gamecenter.GameCenterUIService, "577.8", "Game Center 用户界面服务"

```

```

com.apple.Health, "1.0", "健康"
com.apple.HealthPrivacyService, "1.0", "HealthPrivacyService"
com.apple.Home.HomeUIService, "1", "HomeUIService"
com.apple.InCallService, "1.0", "InCallService"
com.apple.Magnifier, "1", "放大器"
com.apple.MailCompositionService, "3445.104.12", "MailCompositionService"
com.apple.mobilesms.compose, "1.0", "MessagesViewService"
com.apple.mobilephone, "36", "电话"
com.apple.MobileSMS, "1.0", "信息"
com.apple.mobilesafari, "604.1", "Safari 浏览器"
com.apple.mobileslideshow, "43", "照片"
com.apple.mobiletimer, "1.0", "时钟"
com.apple.MusicUIService, "1.0", "MusicUIService"
com.apple.Passbook, "1.0", "钱包"
com.apple.PassbookStub, "1", "Wallet"
com.apple.PassbookUIService, "1.0", "钱包"
com.apple.PhotosViewService, "1", "PhotosViewService"
com.apple.PreBoard, "1.0", "PreBoard"
com.apple.Preferences, "1.0", "设置"
com.apple.PrintKit.Print-Center, "1.0", "打印中心"
com.apple.SIMSetupUIService, "1", "SIMSetupUIService"
com.apple.social.SLYahooAuth, "1", "SLYahooAuth"
com.apple.SafariViewService, "1.0", "SafariViewService"
com.apple.ScreenSharingViewService, "46.2", "ScreenSharingViewService"
com.apple.ScreenshotServicesService, "1", "ScreenshotServicesService"
com.apple.purplebuddy, "1.0", "设置"
com.apple.SharedWebCredentialViewService, "1.0", "SharedWebCredentialViewService"
com.apple.SharingViewService, "1288.66", "SharingViewService"
com.apple.SiriViewService, "1.0", "Siri"
com.apple.susuiservice, "1", "SoftwareUpdateUIService"
com.apple.StoreDemoViewService, "1", "StoreDemoViewService"
com.apple.ios.StoreKitUIService, "1", "iTunes"
com.apple.TVAccessViewService, "282.60.36", "TVAccessViewService"
com.apple.TVRemoteUIService, "1", "TVRemoteUIService"
com.apple.TrustMe, "1.0", "TrustMe"
com.apple.VSViewService, "291.60.3", "VideoSubscriberAccountViewService"
com.apple.webapp, "1.0", "Web"
com.apple.webapp1, "1.0", "WebApp1"
com.apple.WebContentFilter.remoteUI.WebContentAnalysisUI, "1.0", "WebContentAnalysisUI"
com.apple.WebSheet, "1.0", "WebSheet"
com.apple.iad.iAdOptOut, "1.0", "Workbench Ad Tester"
com.apple.CloudKit.ShareBear, "1", "icloud"
kjc.loader, "1", "checkrain"
com.saurik.Cydia, "0.9", "Cydia"
com.tigisoftware.Filza, "2", "Filza"
com.apple.MobileReplayer, "99.97", "MobileReplayer"

```

另外一个已越狱的iPhone7

```

→ ~ ideviceinstaller -l -o list_user
CFBundleIdentifier, CFBundleVersion, CFBundleDisplayName
com.ccb.ccbDemo, "20211207", "中国建设银行"
com.hxb.mobile.client, "0.0.0.6", "华夏银行"
com.baidu.searchcraft, "3.9.0.0", "简单搜索"
com.netease.cloudmusic, "2546", "网易云音乐"

```

```
com.yum.kfc.brand, "1", "肯德基"
com.autohome, "353135", "汽车之家"
com.suzhoubank.SZBank, "5.1.7.2", "苏州银行"
com.cmbchina.MPBBank, "10001", "招商银行"
com.boc.BOCMBCI, "3.0.4", "中国银行"
rn.notes.best, "11122019", "爱思极速版"
com.tencent.xin, "8.0.14.34", "微信"
com.ss.iphone.ugc.Aweme, "178022", "抖音"
science.xnu.undecimus.LT938RA3P2, "1", "unc0ver"
com.crifan.ShowSysInfo, "1", "ShowSysInfo"
```

crifan.org, 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:
2022-11-02 21:40:52

libimobiledevice

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

其中常用的一些工具有：

- `idevice_id`
- `ideviceinfo`

子工具

idevicename

举例：

```
iDeviceName
iPhone7_1331
```

iDeviceDate

举例：

```
iDeviceDate
Wed Nov 2 21:30:01 CST 2022
```

iDeviceProvision

help语法：

```
iDeviceProvision
Usage: iDeviceProvision [OPTIONS] COMMAND

Manage provisioning profiles on a device.

Where COMMAND is one of:
install FILE      Installs the provisioning profile specified by FILE.
                  A valid .mobileprovision file is expected.
list              Get a list of all provisioning profiles on the device.
copy PATH         Retrieves all provisioning profiles from the device and
                  stores them into the existing directory specified by PATH.
                  The files will be stored as UUID.mobileprovision
copy UUID PATH   Retrieves the provisioning profile identified by UUID
                  from the device and stores it into the existing directory
                  specified by PATH. The file will be stored as UUID.mobileprovision.
remove UUID       Removes the provisioning profile identified by UUID.
remove-all        Removes all installed provisioning profiles.
dump FILE         Prints detailed information about the provisioning profile
                  specified by FILE.

The following OPTIONS are accepted:
-u, --udid UDID  target specific device by UDID
-n, --network     connect to network device
-x, --xml         print XML output when using the 'dump' command
-d, --debug       enable communication debugging
-h, --help        prints usage information
-v, --version     prints version information

Homepage: <https://libimobiledevice.org>
Bug Reports: <https://github.com/libimobiledevice/libimobiledevice/issues>
```


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
```

其中会包含很多工具，其中就有此处的 idevice_id

使用

查看已连接iOS设备（往往是iPhone）的ID

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
```

help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
```

ideviceinfo

可以用 `ideviceinfo` 查看当前已连接的iOS设备的各种信息。

安装

```
brew install libimobiledevice
```

其中会包含很多工具，其中就有此处的 `ideviceinfo`

使用

查看设备的其他参数信息：

- `ideviceinfo -u {iOSDeviceId} -k DeviceName`
- `ideviceinfo -u {iOSDeviceId} -k ProductVersion`
- `ideviceinfo -u {iOSDeviceId} -k ProductType`
- `ideviceinfo -u {iOSDeviceId} -k ProductName`

举例：

```
→ ~ ideviceinfo -u ed94089f3e34d5538065a695bfd03dfbb3c5579 -k DeviceName
CriFan iPhone6
→ ~ ideviceinfo -u ed94089f3e34d5538065a695bfd03dfbb3c5579 -k ProductVersion
12.4.5
→ ~ ideviceinfo -u ed94089f3e34d5538065a695bfd03dfbb3c5579 -k ProductType
iPhone7,2
→ ~ ideviceinfo -u ed94089f3e34d5538065a695bfd03dfbb3c5579 -k ProductName
iPhone OS
```

不加参数，则输出全部信息：

举例：

```
ideviceinfo
ActivationState: Activated
BasebandActivationTicketVersion: V2
BasebandCertId: 2315222105
BasebandChipID: 9781473
BasebandKeyHashInformation:
AKeyStatus: 2
SKeyHash: u+/tcCwvaQ+1Y9t40I4yegCEmB28mA1laR0haIVGBWo
SKeyStatus: 0
BasebandMasterKeyHash: 8CB15EE4C8002199070D9500BB8FB183B02713A5CA2A6B92DB5E75CE15536182
BasebandRegionSKU: AAAAAAAAAAAAAAAAAAAAAAAA=AAAAAAAAAAAAAAA=AAAAAAAAAAAAAAA=AAAAAAAAAAAAAAA=
BasebandSerialNumber: yuMYKA=-
BasebandStatus: BBInfoAvailable
```

```
BasebandVersion: 7.30.01
BluetoothAddress: 14:bd:61:9d:3d:01
BoardId: 8
BrickState: false
BuildVersion: 17D50
CPUArchitecture: arm64
CarrierBundleInfoArray[1]:
  0:
    CFBundleIdentifier: com.apple.ChinaTelecom_USIM_cn
    CFBundleVersion: 40.0.1
    IntegratedCircuitCardIdentity: 89860321945128778023
    InternationalMobileSubscriberIdentity: 460115142016109
    MCC: 460
    MNC: 11
    MobileEquipmentIdentifier: 35534108315977
    Slot: kOne
    kCTPostponementInfoAvailable: SIMCarrierInfo
CertID: 2315222105
ChipID: 32784
ChipSerialNo: yuMYKA==
DeviceClass: iPhone
DeviceColor: 2
DeviceName: iPhone7_1331
DieID: 1707430441053478
EthernetAddress: 14:bd:61:9d:3d:02
FirmwareVersion: iBoot-5540.80.2
FusingStatus: 3
HardwareModel: D10AP
HardwarePlatform: t8010
HasSiDP: true
HostAttached: true
IntegratedCircuitCardIdentity: 89860321945128778023
InternationalMobileEquipmentIdentity: 355341083159771
InternationalMobileSubscriberIdentity: 460115142016109
InternationalMobileSubscriberIdentityOverride: false
MLBSerialNumber: F3X65340DPJH6TGD
MobileEquipmentIdentifier: 35534108315977
MobileSubscriberCountryCode: 460
MobileSubscriberNetworkCode: 11
ModelNumber: MNH12
NonVolatileRAM:
  IONVRAM-SYNCNOW-PROPERTY: SU90V1JBT51TWU5DTk9XLVBST1BFU1RZ
  auto-boot: dHJ1ZQ==
  backlight-level: MTQ5NA==
  boot-args:
    com.apple.System.tz0-size: MHhDMDAwMDA=
  oblit-begins: T2JsaXRUeXB1O1BPYmxpdGvYXR1RGF0YVBhcRpdG1vb14gUmVhc29uO1B1bmtub3du
  obliteration: aGFuZGxlX21lc3NhZ2U6IE9ibGl0ZXJhdGlvb1BDb21wbGV0ZQo=
PRIVersion_Major: 0
PRIVersion_Minor: 1
PRIVersion_ReleaseNo: 192
PartitionType: GUID_partition_scheme
PasswordProtected: false
PhoneNumber: +86 xxx xxxx xxxx
PkHash: if98WU1oTt+jcaZ1vUC9AOG/r1bJxZpIN+cg7Rv1bCY
ProductName: iPhone OS
```

```

ProductType: iPhone9,1
ProductVersion: 13.3.1
ProductionSOC: true
ProtocolVersion: 2
ProximitySensorCalibration: f3oABTIAHgC34qNBL26oQQAAyELcV5BB8x98QQAAFkPcV5BB8x98QQAAFkPyMktDG2R
YwB9thUQAkIVE4fqERK5nhESCc91BppveQUdy8kEOrfpBAAgQhkEAEM60gtCAABIQg00skMbr7FD20oDRt9vVkoGGB4eAi
AYDvRx1kXj1Y1DSDmuRQIAAAAAACa
RegionInfo: CH/A
SIMStatus: kCTSimsupportSIMstatusReady
SIMTrayStatus: kCTSimsupportSIMtrayInsertedWithSIM
SerialNumber: DNPT1P8EHG74
SoftwareBehavior: QQAaaaaaaaaaaaaaa==
SoftwareBundleVersion:
SupportedDeviceFamilies[1]:
  0: 1
TelephonyCapability: true
TimeIntervalSince1970: 1667395772.152844
TimeZone: Asia/Shanghai
TimeZoneOffsetFromUTC: 28800.000000
TrustedHostAttached: true
UniqueChipID: 1707430441053478
UniqueDeviceID: adab53e3250e8be1ee0db75bccdc2063df608b46
UseRaptorCerts: true
Uses24HourClock: true
WiFiAddress: 14:bd:61:9d:3d:00
WirelessBoardSerialNumber: F5007716F2
kCTPostponementInfoPRIVersion: 0.1.192
kCTPostponementInfoPRLName: 301
kCTPostponementInfoServiceProvisioningState: true
kCTPostponementStatus: kCTPostponementStatusActivated

```

help语法

```

ideviceinfo --help
Usage: ideviceinfo [OPTIONS]

Show information about a connected device.

OPTIONS:
  -u, --udid UDID      target specific device by UDID
  -n, --network        connect to network device
  -s, --simple         use a simple connection to avoid auto-pairing with the device
  -q, --domain NAME   set domain of query to NAME. Default: None
  -k, --key NAME       only query key specified by NAME. Default: All keys.
  -x, --xml            output information as xml plist instead of key/value pairs
  -h, --help           prints usage information
  -d, --debug          enable communication debugging
  -v, --version        prints version information

Known domains are:

  com.apple.disk_usage
  com.apple.disk_usage.factory
  com.apple.mobile.battery

```

```
com.apple.iqagent
com.apple.purplebuddy
com.apple.PurpleBuddy
com.apple.mobile.chaperone
com.apple.mobile.third_party_termination
com.apple.mobile.lockdownd
com.apple.mobile.lockdown_cache
com.apple.xcode.developerdomain
com.apple.international
com.apple.mobile.data_sync
com.apple.mobile.tethered_sync
com.apple.mobile.mobile_application_usage
com.apple.mobile.backup
com.apple.mobile.nikita
com.apple.mobile.restriction
com.apple.mobile.user_preferences
com.apple.mobile.sync_data_class
com.apple.mobile.software_behavior
com.apple.mobile.iTunes.SQLMusiclibraryPostProcessCommands
com.apple.mobile.iTunes.accessories
com.apple.mobile.internal
com.apple.mobile.wireless_lockdown
com.apple.fairplay
com.apple.iTunes
com.apple.mobile.iTunes.store
com.apple.mobile.iTunes
```

Homepage: <<https://libimobiledevice.org>>

Bug Reports: <<https://github.com/libimobiledevice/libimobiledevice/issues>>

端口转发

用 iproxy 或 mobiledevice

- 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:

- 省略设备ID

```
iproxy 8100 8100
```

- 当前只连接一个iOS设备时, 可以不指定, 忽略 UDID

- 指定设备ID

```
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)
```

- mobiledevice

- 安装:

```
brew install mobiledevice
```

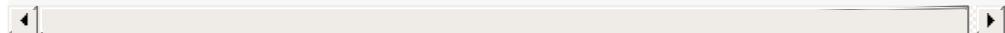
- 用法:

- 省略设备ID

```
mobiledevice tunnel 8100 8100
```

- 指定设备ID

```
mobiledevice 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]
```

mobiledevice

安装：

```
brew install mobiledevice
```

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)

```

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: IOCreatePlugInInterfaceForService failed 0xe00002be
2020-04-29 14:50:16.086 system_profiler[46290:995081] SPUSBDevice: IOCreatePlugInInterfaceForService failed 0xe00002be
2020-04-29 14:50:16.087 system_profiler[46290:995081] SPUSBDevice: IOCreatePlugInInterfaceForService failed 0xe00002be
2020-04-29 14:50:16.087 system_profiler[46290:995081] SPUSBDevice: IOCreatePlugInInterfaceForService failed 0xe00002be
2020-04-29 14:50:16.088 system_profiler[46290:995081] SPUSBDevice: IOCreatePlugInInterfaceForService failed 0xe00002be
2020-04-29 14:50:16.089 system_profiler[46290:995081] SPUSBDevice: IOCreatePlugInInterfaceForService 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.org, 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-2F5C74E
404CA] (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"
```


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.org, 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.org, 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自动化测试相关

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>

Search

mainScreen

All Reference Guides Sample Code Videos

7 results for "mainScreen"

mainScreen

UIKit > UIScreen > mainScreen

Returns the screen object representing the device's screen.

iOS MAC CATALYST tvOS Objective-C

mainScreen

AppKit > NSScreen > mainScreen

Returns the screen object containing the window with the keyboard focus.

macOS Objective-C

mainScreen

XCTest > XCUIScreen > mainScreen

The current device's main screen.

XCODE Objective-C

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];
```

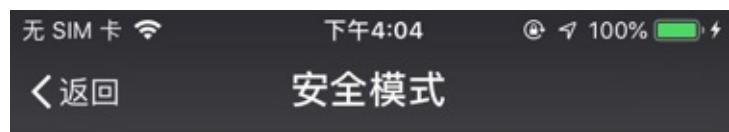
才可以。

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：





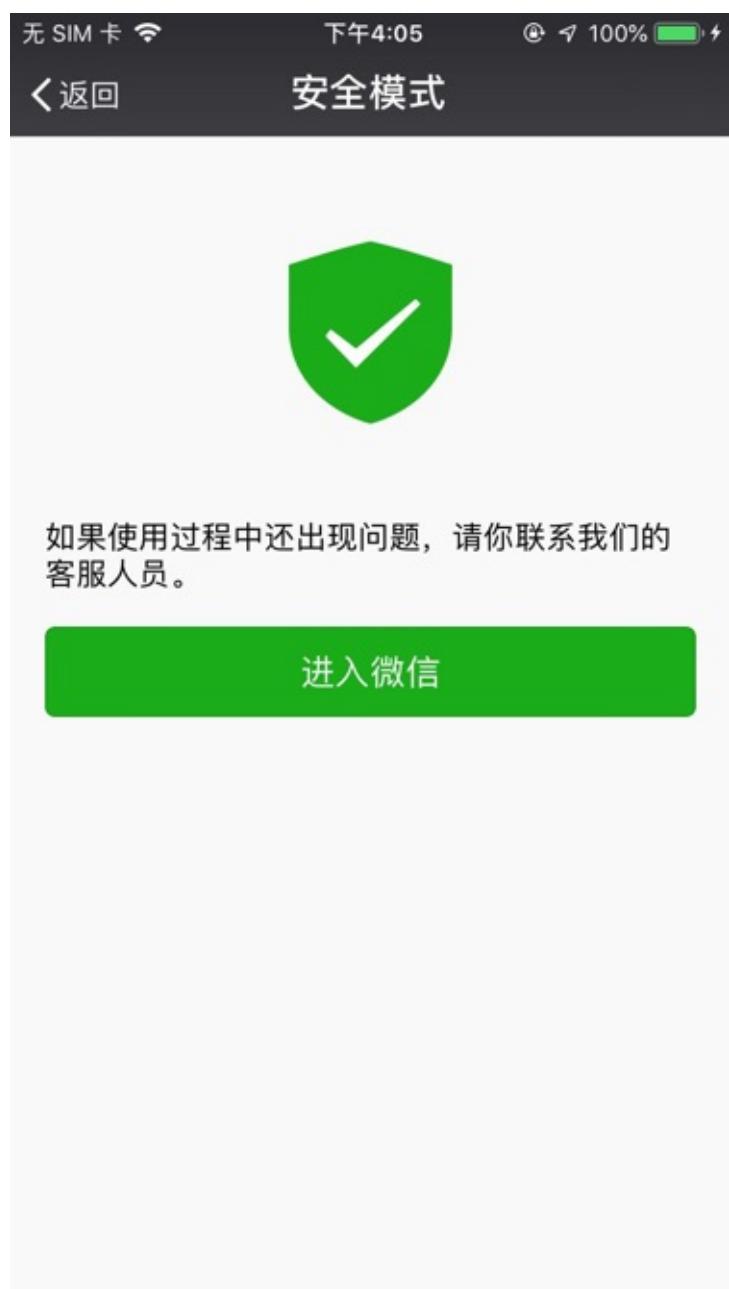
清理缓存会清理你的手机本地缓存文件，但不会清理你的消息数据，使用后需要重新登录微信

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.org, 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：
2022-11-03 11:26:06

iPhone

TODO:

- 【已解决】已越狱iPhone的iOS中如何命令行创建一个大文件
- 【记录】iPhone手机恢复还原出厂设置
- 【已解决】iPhone中查看ARM版本: armv6、armv7
-
- 设备识别
 - 【已解决】iPhone插入Mac后提示: 可能不支持此配件
 - 【已解决】Mac无法识别插入的已越狱iPhone7
- 状态栏
 - 【已解决】iPhone中左上角状态栏中显示运营商的效果
 - 【已解决】iPhone顶部状态栏中图标和信息的具体含义
- app管理
 - 【已解决】Mac中如何获取到iPhone中自带应用设置的bundle ID
 - 【无法解决】越狱iPhone中把iOS的app移动到/Applications目录
- 卡贴机
 - 【已解决】iPhone卡贴机
 - 【整理】购买二手iPhone测试机
- 文件管理
 - 【已解决】已越狱iOS中如何通过ssh从iPhone拷贝出文件
- SSH
 - 【已解决】给已越狱iPhone6修改ssh密码并开启ssh免密登录
 - 【记录】给越狱iPhone7安装SSH并配置免密登录
 - 【已解决】ssh登录iPhone失败: Host key verification failed
- SSH证书
 - 【已解决】Mac中删除已有ssh证书和已安装到iPhone中的ssh证书

ios 开发期间，用到的最多的设备就是 iPhone 了，此处整理iPhone的相关开发心得。

iPhone的类型和尺寸

苹果的移动端的设备，典型都有 iPhone , iPad 等，所用的操作系统都是iOS。

其中用的最多是 iPhone 。

iPhone 的设备类型，相对较固定，数量有限。

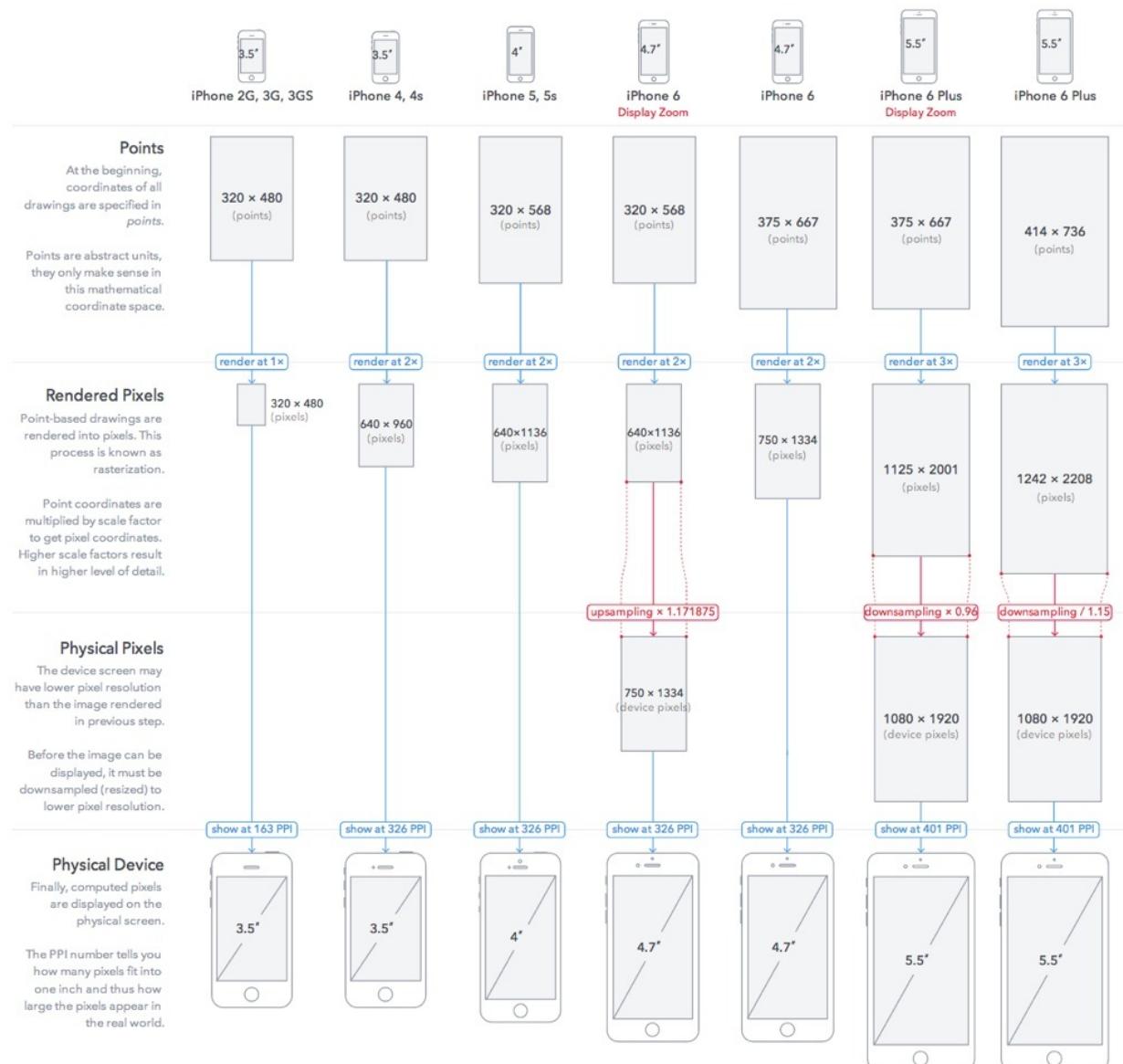
下面总结一下，不同的iPhone设备的屏幕相关的参数：

机型	屏幕宽高/单位点	屏幕模式	屏幕对角线长度
iPhone 3GS	320x480	1x	3.5英寸
iPhone 4	320x480	2x	3.5英寸

iPhone 5	320x568	2x	4英寸
iPhone 6	375x667	2x	4.7英寸
iPhone 6 Plus	414x736	3x	5.5英寸

对应的屏幕类型分辨率和像素的换算关系为：

The Ultimate Guide To iPhone Resolutions



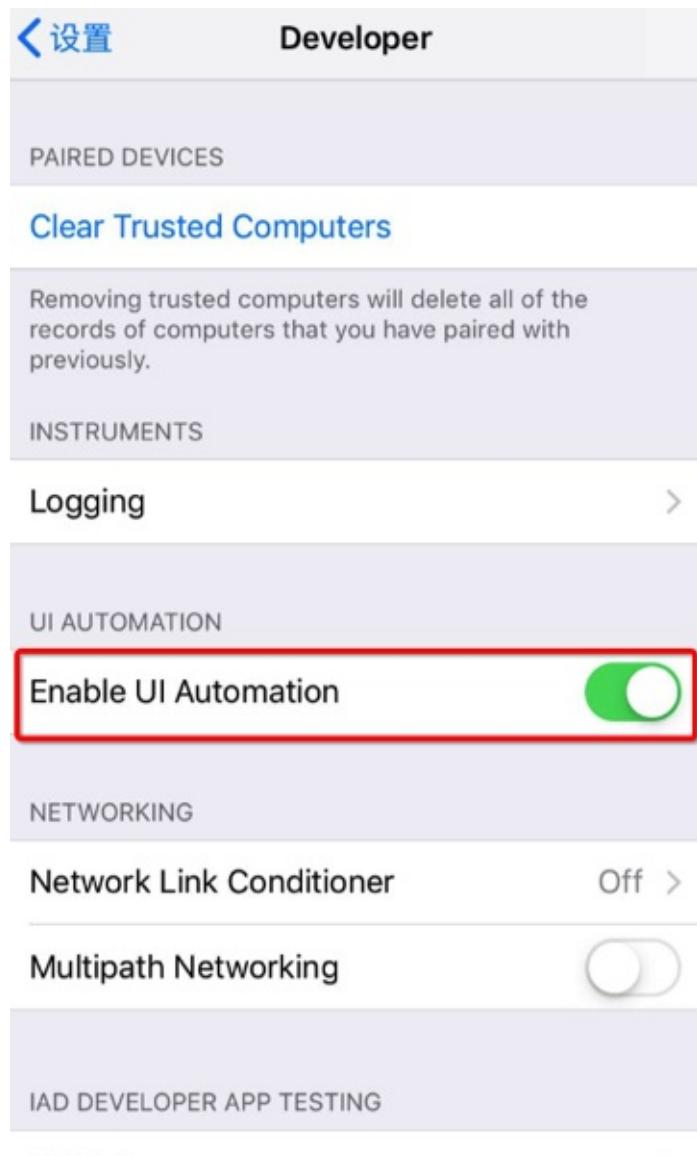
开发相关设置

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

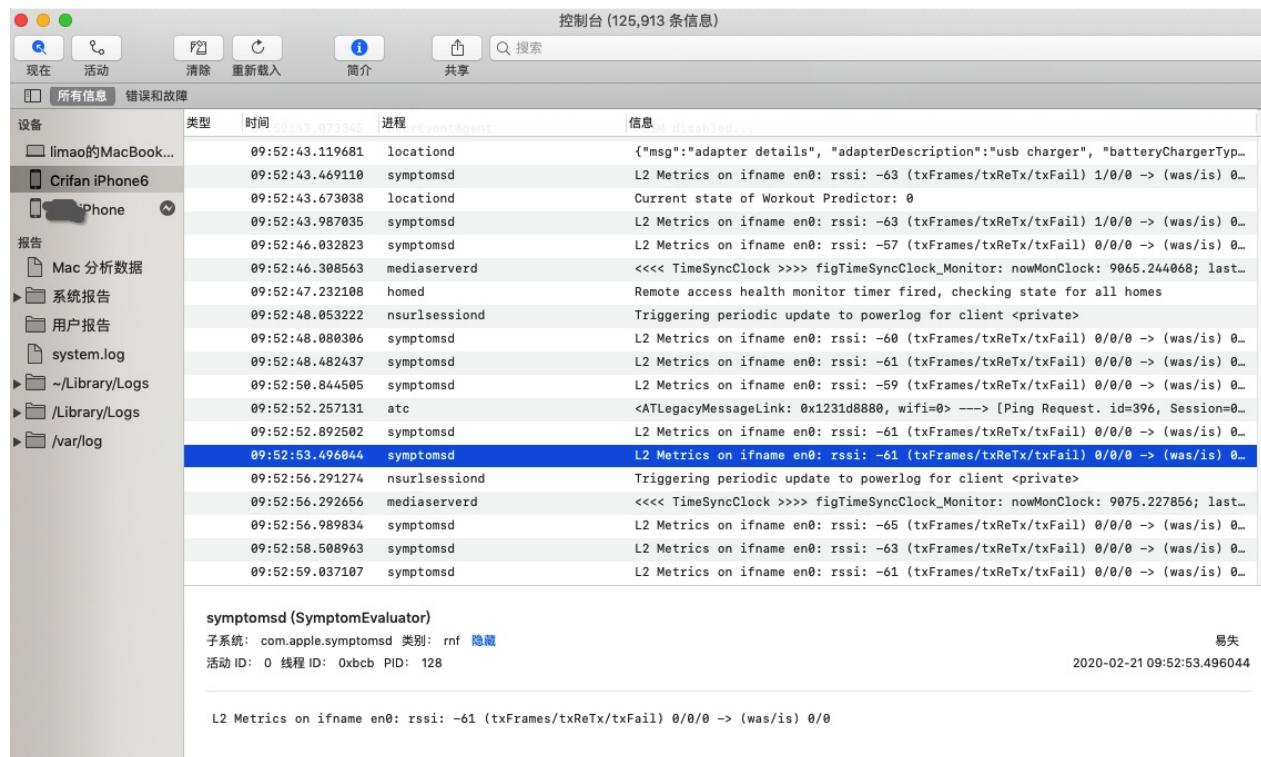
如何打开控制台：

- 启动台Launch Pad -> 其他 -> 控制台

- - Xcode->Window->Devices and Simulators->Devices

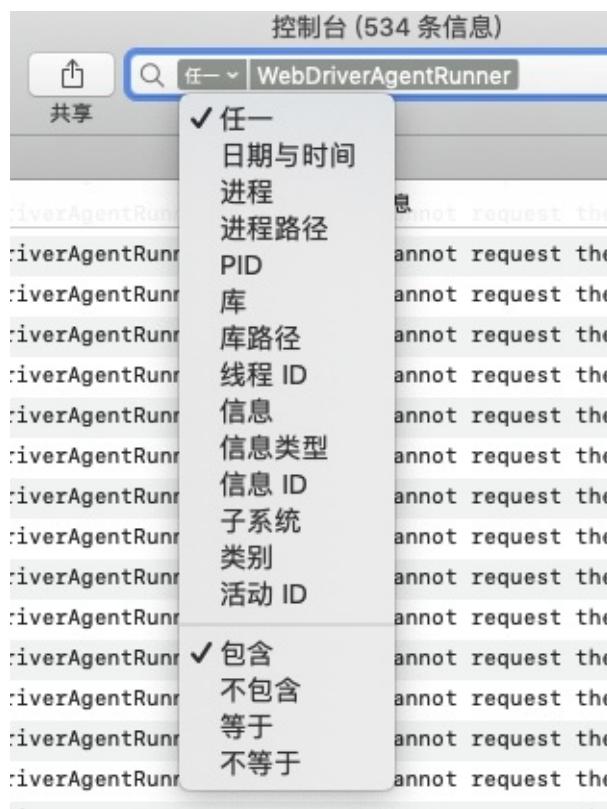
◦

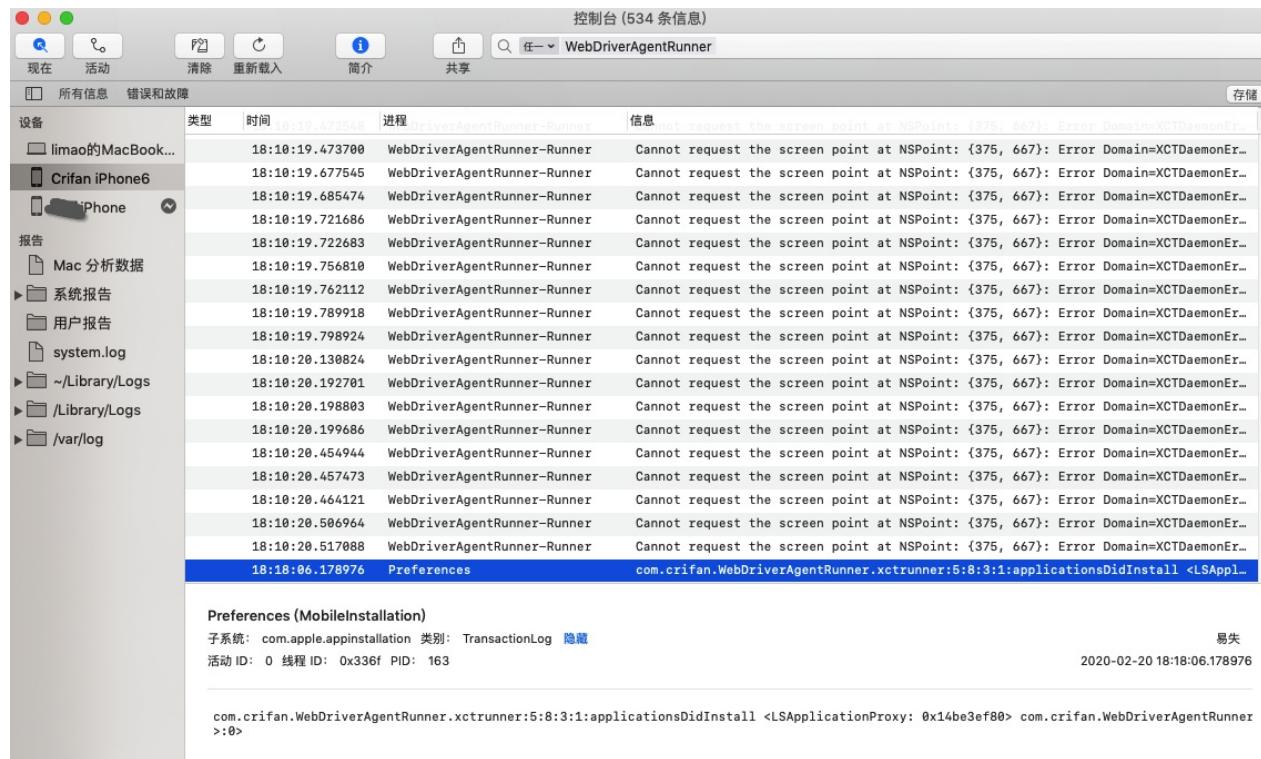
启动后，即可看到iPhone真机的log日志了：



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.org, 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:

2022-11-03 17:29:15

子教程

和Apple苹果相关的内容比较多。现已把其他独立的部分，整理出对应子教程：

- [iOS开发心得](#)
 - [iOS自动化测试](#)
 - [iOS自动化测试利器：facebook-wda](#)
 - [相关（父）教程](#)
 - [移动端自动化测试概览](#)
 - [相关（父）教程](#)
 - [移动端APP开发总结 crifan.org](#)
- [Xcode开发心得](#)
 - [Xcode开发：调试心得 \(crifan.org\)](#)
 - [Xcode内置调试器：LLDB \(crifan.org\)](#)

crifan.org, 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：
2022-11-03 17:38:38

附录

下面列出相关参考资料。

crifan.org, 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 website for the UIScreen class. The URL in the browser is developer.apple.com/documentation/uikit/uiscreen. The page title is 'UIScreen'. The top navigation bar includes links for Documentation, 4.4.0 文档 — B..., openat://faceb..., /wda/screen -..., Displays, and UIScreen - UIKit...+. The 'Discover' tab is selected. The main content area shows the 'Class' section for UIScreen. It includes the declaration:

```
class UIScreen : NSObject
```

The 'Overview' section states: "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". On the right side, there is a sidebar with sections for SDKs (iOS 2.0+, Mac Catalyst 13.0+, tvOS 9.0+), Framework (UIKit), and 'On This Page' links (Declaration, Overview, Topics, Relationships, See Also).

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

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.org, 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
- Objective-C 的第三方 IDE AppCode 怎么样? - 知乎
-

crifan.org, 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：
2022-11-03 09:46:41