

flutter 项目本身

所需开发工具环境

flutter 谷歌公司开发 系统支持库 镜像库

flutter 官网：

<https://flutter.dev/community/china>

// 步骤 1 .bash_profile

touch .bash_profile

pwd

/Users/haijunyan

open ~

export PUB_HOSTED_URL=https://pub.flutter-io.cn

export FLUTTER_STORAGE_BASE_URL=<https://storage.flutter-io.cn>

// 步骤 2 获取 Flutter SDK

<https://docs.flutter.dev/release/archive>

mac OS Stable channel (macOS) 3106

flutter_macos_3.10.6-stable.zip

(支持) 系统 flutter 底层文件夹 (安装包) 至目标目录

/Users/haijunyan/Documents

/Users/haijunyan/Documents/flutter

系统 flutter 底层文件夹 (安装包) 的访问路径配置至全局变量配置文件

export PATH=/Users/haijunyan/Documents/flutter/bin:\$PATH

//FlutterSDK 支持开发环境检验

flutter doctor

交互式脚本 (命令行交互区)

报错: zsh: command not found: flutter

Mac 默认采用 zsh 交互式脚本 (命令行交互区) 作为 登录 Shell / 交互式 Shell

解决方案一: 手动切换至 bash 交互式脚本 (命令行交互区)

source ~/.bash_profile // source /Users/haijunyan/.bash_profile

flutter doctor

解决方案二:

pwd // /Users/haijunyan

cd /Users/haijunyan/

open ~/.zshrc

vim ~/.zshrc //新建文件 (若提示文件不存在)

open ~/.bash_profile

把 bash_profile 中的内容 copy 到 zshrc 文件中 (全局变量配置)

source ~/.zshrc //再资源载入 source 命令重新加载

终端退出重启 flutter doctor

AndroidStudio Google 公司提供的兼容性辅助工具 可构建出 android/ios/window/macOS/linux 等兼容各个方向的运行包

AndroidStudio (version 最新 Mac)

右上角 更多

Apply > OK

flutter doctor --android-licenses //y

AndroidStudio 兼容性工具平台的配置

.bash_profile

Xcode 专业辅助构建出 iOS 运行包

依托于 flutter 项目 产生的 app 应用包

创建基(本)层的 flutter 项目

cd /Users/haijunyan/Desktop/CustomKit/TeamGroupSimulation/FlutterDemo/

source /Users/haijunyan/.bash_profile 或 source ~/.bash_profile

flutter create 项目文件夹名称

运行基层的 flutter 项目

切换至 flutter 项目对应的文件夹路径下

cd 相对路径 //cd fltapp

正式运行基层的 flutter 项目

flutter run //命令行运行 (手动关联模拟器或手机设备)

brew -v //判断是否安装 Homebrew 环境

brew update //brew 环境更新至最新版本

//安装辅助插件 (flutter 项目 iOS 包 运行 iOS 真机)

xcode-select --install //Xcode 相关 工具命令行

//安装 ipa 包 (通过命令码实现 (需要预配置好开源库插件环境)(不推荐))

brew cleanup

brew install --HEAD usbmuxd

Error: libusbmuxd 2.0.2 is already installed

brew unlink libusbmuxd

brew install --HEAD usbmuxd //brew update --auto-update

brew link usbmuxd

brew install --HEAD libimobiledevice

brew unlink imobiledevice

brew install --HEAD imobiledevice //强调分支下载

brew install libimobiledevice //***不强调分支 (基于 tag 下载)***

brew link libimobiledevice

brew install ideviceinstaller ios-deploy cocoapods

pod setup

//安装 lipa 包 (通过辅助工具实现 (itools/iTunes/Xcode 等工具平台) (推荐))

brew doctor

切换至二方库 flutter 项目本身的路径目录文件夹下

cd /Users/haijunyan/Desktop/FlutterDemo/fltapp

pwd

/Users/haijunyan/Desktop/FlutterDemo/fltapp

open ios/Runner.xcworkspace //ios 方向 workspace 运行研究 (默认打开 Xcode 的 workspace 编辑模式)

AndroidStudio 兼容性工具平台 (推荐 (自动关联))

左 模拟器

右 真机

热重载 r 更改的部分刷新 维持状态

热重启 R 全部刷新 全部刷新重启

二方库 flutter 项目 通过模拟器运行自动热重载 (实时更新 (调试推荐))

二方库 flutter 项目 通过真机运行需要手动热重载 (手动更新 (调试不推荐))

备注:

flutter 业务层面 lib 业务 源码资源

二方库 flutter 项目 (本身) pubspec 库本身配置文件

iOS 宿主层面 (iOS 体系 (iPhone iPod iWatch iPad ITV))

android 宿主层面 (Android 体系 (安卓手机 安卓手环 (表) 安卓平板 安卓电视 安卓 GoogleChrome 浏览器))

macos 宿主层面 Mac 电脑端原生项目产生电脑.dmp 产品包 (mac 电脑设备)

windows 宿主层面 window 电脑端原生项目产生电脑.exe 产品包 (window 电脑设备)

linux 宿主层面 linux 电脑端原生项目产生电脑产品包 (linux 电脑设备)

*****二方库 flutter 本身源码 +xxx(iOS/macos/Android/windows/linux) 原生项目 (包)*****

*****基于硬件的待渲染载体设备触发对应的 xxx 原生项目 (包)*****

flutter 运行流程之 iOS 方向 关键要素 2 次中转

Xcode 工具平台 iOS 方向产品包 对应 iOS(真机/模拟器) 设备 运行视觉流程效果

AndroidStudio 工具平台 兼容型各方向的产品包 对应各种兼容性(真机/模拟器) 设备 运行视觉流程效果

****二方库 flutter 项目通过中转式运行后最终实际产生的目标程序为原生移动端 iOS(Xcode 工具平台)/Android(AndroidStudio 工具平台) 产品包****

keg 文件夹

formulae 法则/手段

brew 基层管理环境(管理上层插件) //通过 brew 环境下载插件的过程非常缓慢(大概率断开失效, 不断重新连接网络重新下载)

1.辅助工具实现安装 ipa(推荐)

2.命令码实现安装 ipa(不推荐)

libimobiledevice 本质是开源库 (-HEAD/-MASTER ...强调各种分支 branch)

pod --version

辅助工具 AndroidStudio 工具平台编辑模式: 可开发 Android 原生项目产生安卓.apk 产品包

二方库 flutter 项目本身层面自有文件的变更通过 AndroidStudio 工具平台

AndroidStudio 工具平台: 兼容型原生项目层文件的变更+flutter 项目层自有文件的变更

Xcode 工具平台: 原生 iOS 项目层文件的变更

AndroidStudio 工具平台: 兼容各种原生项目(不同运行平台的项目框架(器))

原生项目包(通常比较干净)即项目框架(器)(承载体作用)内被注入的资源通常为业务需求资源(lib 文件夹内的业务源码主文件资源(即 lib 文件夹内进行 flutter 源码开发))

haijunyan 15.7.3(版本高) iPhone7Plus

真机调试:

举例: iPhoneYHJ 12.5.7 iPhone5s

cd /Users/haijunyan/Desktop/FlutterDemo/fltapp

open ios/Runner.xcworkspace

备注:

****iOS 真机调试**:**

AndroidStudio 工具平台 +Xcode 工具平台

1.flutter 项目首次: AndroidStudio 工具平台设备运行>Xcode 工具平台团队签名配置预安装>关闭退出 Xcode 工具平台>AndroidStudio 工具平台设备运行安装真机调试

2.flutter 项目非首次: AndroidStudio 工具平台设备运行安装真机调试

****android 安卓真机调试**:**

AndroidStudio 工具平台

1.flutter 项目: AndroidStudio 工具平台设备运行安装真机调试

mac 和 iphone 数据线连接断续闪烁

解决方案:

终端 Terminal 输入 `sudo killall -STOP -c usbd` 回车

进程关闭 `sudo killall -STOP -c usbd`

安卓开发者官网:

<http://developer.android.com/studio> // 国境外 (VPN)

<https://developer.android.com/studio#downloads>

<https://developer.android.com/studio/run/emulator-acceleration.html>

[https://developer.android.com/studio/run/emulator-acceleration.html#vm-](https://developer.android.com/studio/run/emulator-acceleration.html#vm-mac)

[mac](#) // mac 使用的加速器插件 (境外)

<https://developer.android.google.cn/studio> // 国境内镜像下载站

<https://developer.android.google.cn/studio/intro?hl=zh-cn> // AndroidStudio 官方说明文档

安卓模拟器 处于 AndroidStudio 自有的 Dock 栏区 (window 窗口)

```
source ~/.bash_profile
```

```
emulator -list-avds // 列出 AndroidStudio 工具相关的 android 模拟器设备插件
```

```
emulator -avd Nexus_6P_API_29 // 启动 AndroidStudio 工具相关的 android 模拟器
```

```
emulator -avd Pixel_2_API_29
```

安卓真机数据线连接 (默认调试):

华为荣耀 50NTHAN00 手机设备 (荣耀 50 开发者模式: 设置 > 关于手机 > 版本号 多次连续点击 > 开发者模式)

设置 > 系统和更新 > 开发人员选项 > USB 调试 打开 > 终端 `source ~/.bash_profile` > 终端 `flutter devices` (验证 flutter 项目是否识别真机) >

1. AndroidStudio 工具平台调试: 运行 flutter 项目

2. 终端调试:

```
cd /Users/haijunyan/Desktop/FlutterDemo/fltapp
```

```
pwd
```

```
/Users/haijunyan/Desktop/FlutterDemo/fltapp
```

```
flutter run -d '设备名称' // 运行 flutter 项目
```

安卓真机无线连接调试 WLAN (WIFI 无线调试) (比较慢 / 受到 Wi-Fi 网络质量影响):

设置 > 系统和更新 > 开发人员选项 > USB 调试 打开 > 无线调试 > 使用二维码配对设备 > 已配对的设备 > 无线调试

三方库 flutter 项目引入其他三方开源包 **Package**:

举例: english_words 三方开源包

1. 打开 flutter 项目目录下 pubspec.yaml 文件

2. dependencies 新增 english_words 配置

```
# english_words 三方开源包
```

```
english_words: ^3.1.0
```

- 3.flutter pub get 开源包相关数据资源拖入项目中
- 4.english_words: ^4.0.0 三方开源包采用最新版本
- 5.flutter pub upgrade 重新更新 三方开源包
- 6.重新启动模拟器

flutter/dart相关辅助插件的安装：

AndroidStudio 工具平台 右上角账号旁边设置图标 Plugins Plugins Flutter plugin
搜索辅助插件 install OK

侧边栏 (目录) 项目面板

二方库 插件项目 整体对外 插件感

```
sudo killall -STOP -c usbd  
buzhidao
```