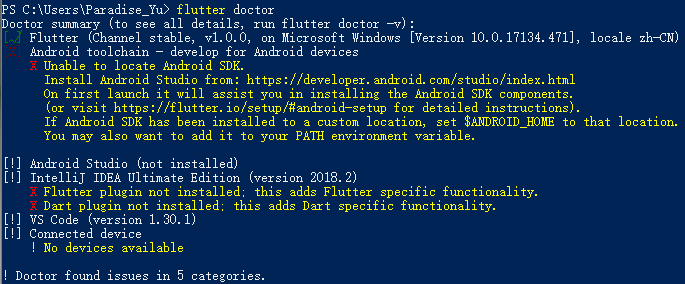
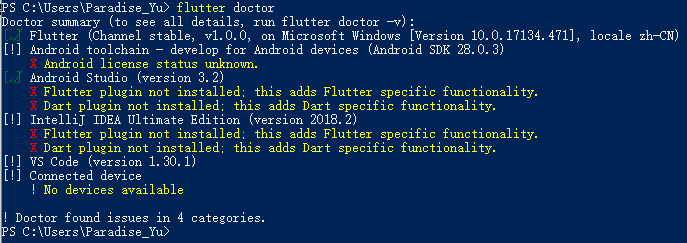
# 2018.12.20 周四

根据官网安装完Flutter之后，首先需要在PowerShell中运行flutter doctor命令，检查Flutter运行所需的组件是否都齐备。我首先出现的界面是：



有三个大红×，一个一个解决。

首先需要安装Android SDK，根据图上的提示安装Android Studio。我也顺利安装完了。回来继续运行flutter doctor，又出现下图：



第一个问题：Android license status unknown.

解决的办法就是运行命令：flutter doctor --android-licenses

但是还是弹出了一个提示框：

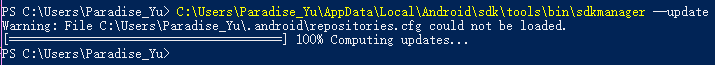


听这意思就是需要更新版本的SDK，先按照提示的命令执行一下看看结果：



提示：Unsupported major.minor version 52.0异常。

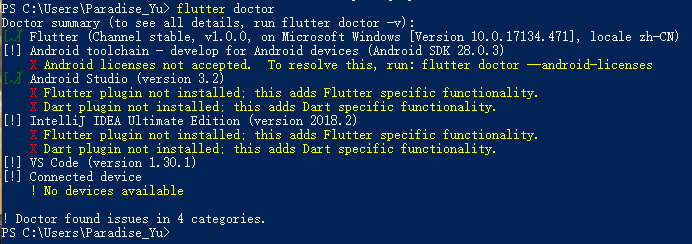
原因是与Java SDK的版本不一致有关，我当前系统采用的JDK版本是1.7，我记得安卓开发至少需要1.8，于是我把系统变量改成1.8。再来试一次：



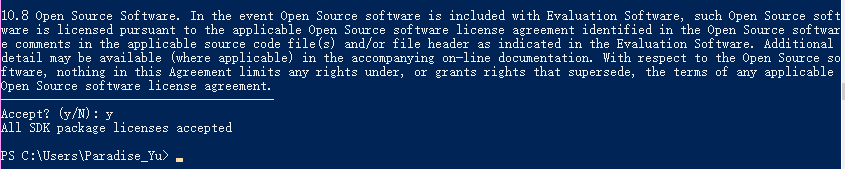
|  |
| --- |
| 2019.7.26 补充：  不一定是因为版本号低的问题导致的这个错误，我的一台电脑安装的是Java9，但是安装位置在D盘，而且jre安装到了jdk的目录下了，也是报错了，我卸载了，又重新安装了8就好了。 |

成功了。不过这是新开的PowerShell窗口，原来的窗口没有监听到JDK的变更，继续运行这个命令还会报同样的错误。

我们再来运行一下doctor：

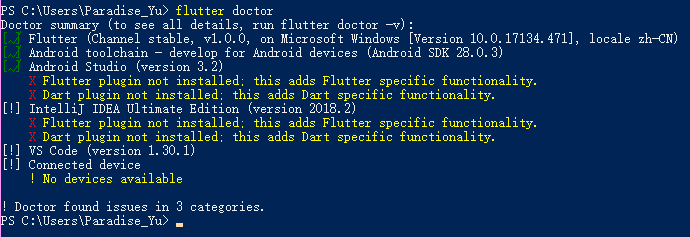


第一个错误的提示内容已经变了，变成了还未接受的安卓信用凭证。并且提示我们用什么命令来解决问题。那我们就照着办就好了：



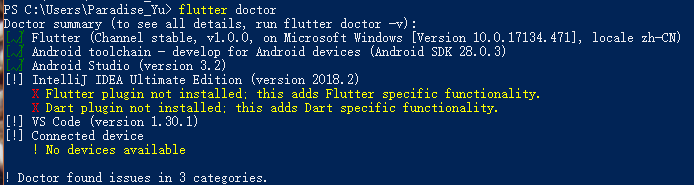
截图只是一小部分，你需要输入多次y来同意协议。很多次。然后才是截图中说的：All SDK package licenses accepted.

再运行flutter doctor，看看还有什么问题：

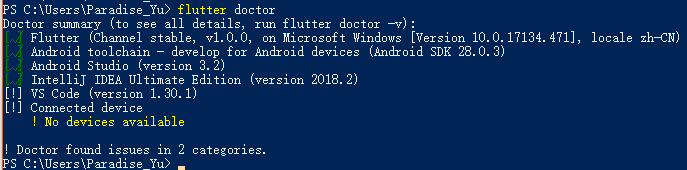


接下来的问题就是Flutter plugin和Dart plugin还未安装在Android Studio中。

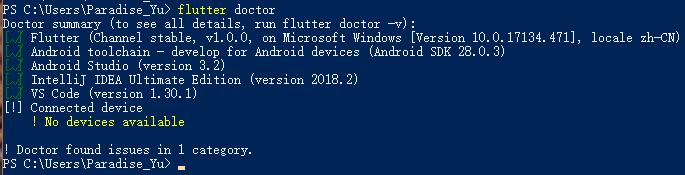
安装的步骤很简单，就是去Android Studio中找Plugin，安装Flutter的同时也把Dart安装好了。然后回来在运行doctor：



好，接下来就是给Intellij idea安装插件。安装方式和Android Studio一模一样，毕竟都是同一家公司的产品。再回来运行doctor：



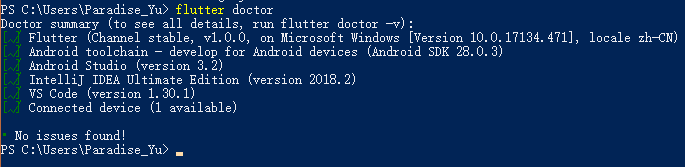
到这里可以说你已经可以用Android Studio或者idea进行开发Flutter了。不过VS Code还不行，你需要到VS Code中去安装Flutter和Dart插件。安装完成后再回来运行doctor：



哎，非常完美了。哦，还有最后一项，那就是这个：No Devices available。这个就是你没连接手机的问题。

只要你把手机设置到开发者模式并且允许USB调试，连到手机上就可以了，我指的是Android手机。目前，我也只考虑开发安卓应用，iOS的再说。

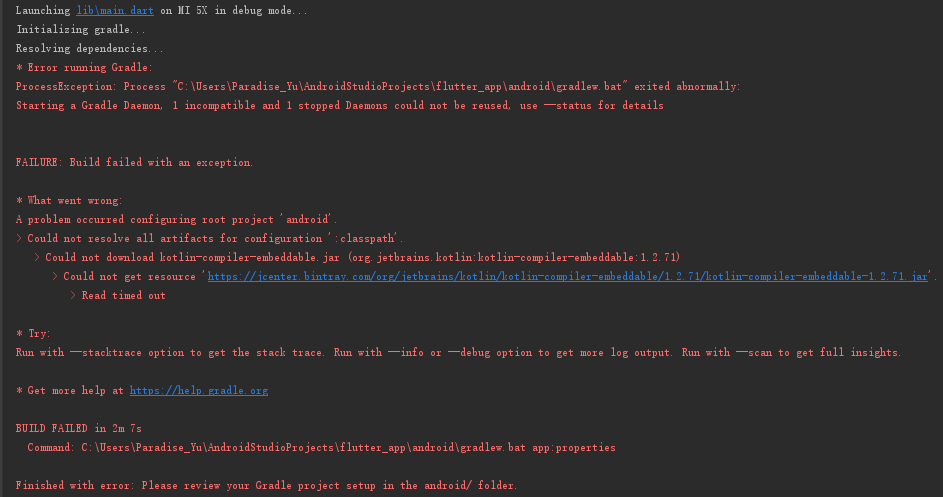
连接手机后应该显示：



自此，安装Flutter就全部完成啦！

# 2018.12.24 周一

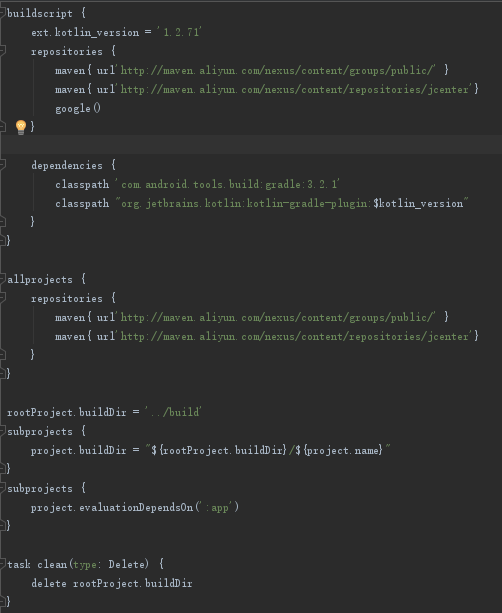
安装完Flutter，接下来就是仪式性的HelloWorld程序了。首先用Android Studio开发第一个Flutter程序。我首先在本地安装了Gradle，版本是4.6。并且配置到了Android Studio中。接下来运行报错了：



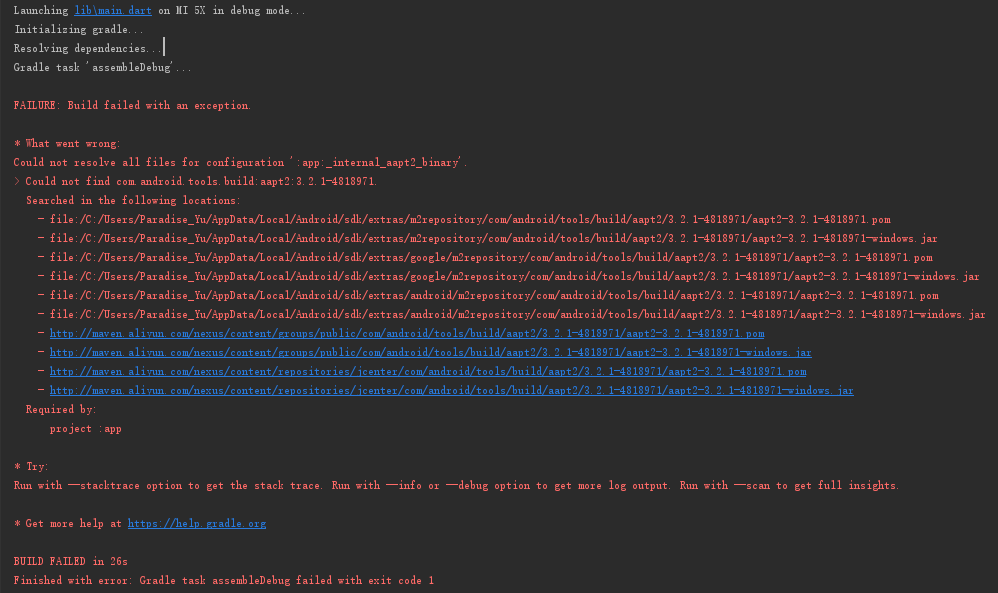
|  |
| --- |
| \* Error running Gradle:  ProcessException: Process "C:\Users\Paradise\_Yu\AndroidStudioProjects\flutter\_app\android\gradlew.bat" exited abnormally:  Starting a Gradle Daemon, 1 incompatible and 1 stopped Daemons could not be reused, use --status for details  FAILURE: Build failed with an exception.  \* What went wrong:  A problem occurred configuring root project 'android'.  > Could not resolve all artifacts for configuration ':classpath'.  > Could not download kotlin-compiler-embeddable.jar (org.jetbrains.kotlin:kotlin-compiler-embeddable:1.2.71)  > Could not get resource 'https://jcenter.bintray.com/org/jetbrains/kotlin/kotlin-compiler-embeddable/1.2.71/kotlin-compiler-embeddable-1.2.71.jar'.  > Read timed out  \* Try:  Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output. Run with --scan to get full insights.  \* Get more help at https://help.gradle.org  BUILD FAILED in 2m 7s  Command: C:\Users\Paradise\_Yu\AndroidStudioProjects\flutter\_app\android\gradlew.bat app:properties  Finished with error: Please review your Gradle project setup in the android/ folder. |

错误的大概意思是有些包下不下来。我在想就是大陆限制访问的问题。于是上网查了查，可以修改repo为阿里云的镜像。

修改build.gradle文件：

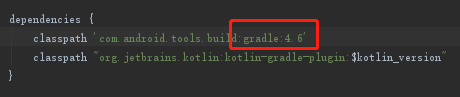


再次运行，又遇到了错误：



|  |
| --- |
| FAILURE: Build failed with an exception.  \* What went wrong:  Could not resolve all files for configuration ':app:\_internal\_aapt2\_binary'.  > Could not find com.android.tools.build:aapt2:3.2.1-4818971.  Searched in the following locations:  - file:/C:/Users/Paradise\_Yu/AppData/Local/Android/sdk/extras/m2repository/com/android/tools/build/aapt2/3.2.1-4818971/aapt2-3.2.1-4818971.pom  - file:/C:/Users/Paradise\_Yu/AppData/Local/Android/sdk/extras/m2repository/com/android/tools/build/aapt2/3.2.1-4818971/aapt2-3.2.1-4818971-windows.jar  - file:/C:/Users/Paradise\_Yu/AppData/Local/Android/sdk/extras/google/m2repository/com/android/tools/build/aapt2/3.2.1-4818971/aapt2-3.2.1-4818971.pom  - file:/C:/Users/Paradise\_Yu/AppData/Local/Android/sdk/extras/google/m2repository/com/android/tools/build/aapt2/3.2.1-4818971/aapt2-3.2.1-4818971-windows.jar  - file:/C:/Users/Paradise\_Yu/AppData/Local/Android/sdk/extras/android/m2repository/com/android/tools/build/aapt2/3.2.1-4818971/aapt2-3.2.1-4818971.pom  - file:/C:/Users/Paradise\_Yu/AppData/Local/Android/sdk/extras/android/m2repository/com/android/tools/build/aapt2/3.2.1-4818971/aapt2-3.2.1-4818971-windows.jar  - http://maven.aliyun.com/nexus/content/groups/public/com/android/tools/build/aapt2/3.2.1-4818971/aapt2-3.2.1-4818971.pom  - http://maven.aliyun.com/nexus/content/groups/public/com/android/tools/build/aapt2/3.2.1-4818971/aapt2-3.2.1-4818971-windows.jar  - http://maven.aliyun.com/nexus/content/repositories/jcenter/com/android/tools/build/aapt2/3.2.1-4818971/aapt2-3.2.1-4818971.pom  - http://maven.aliyun.com/nexus/content/repositories/jcenter/com/android/tools/build/aapt2/3.2.1-4818971/aapt2-3.2.1-4818971-windows.jar  Required by:  project :app  \* Try:  Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output. Run with --scan to get full insights.  \* Get more help at https://help.gradle.org  BUILD FAILED in 26s  Finished with error: Gradle task assembleDebug failed with exit code 1 |

错误中有一串数字：3.2.1，强烈怀疑是因为我安装的gradle是4.6版本，而这个配置文件写的是3.2.1：



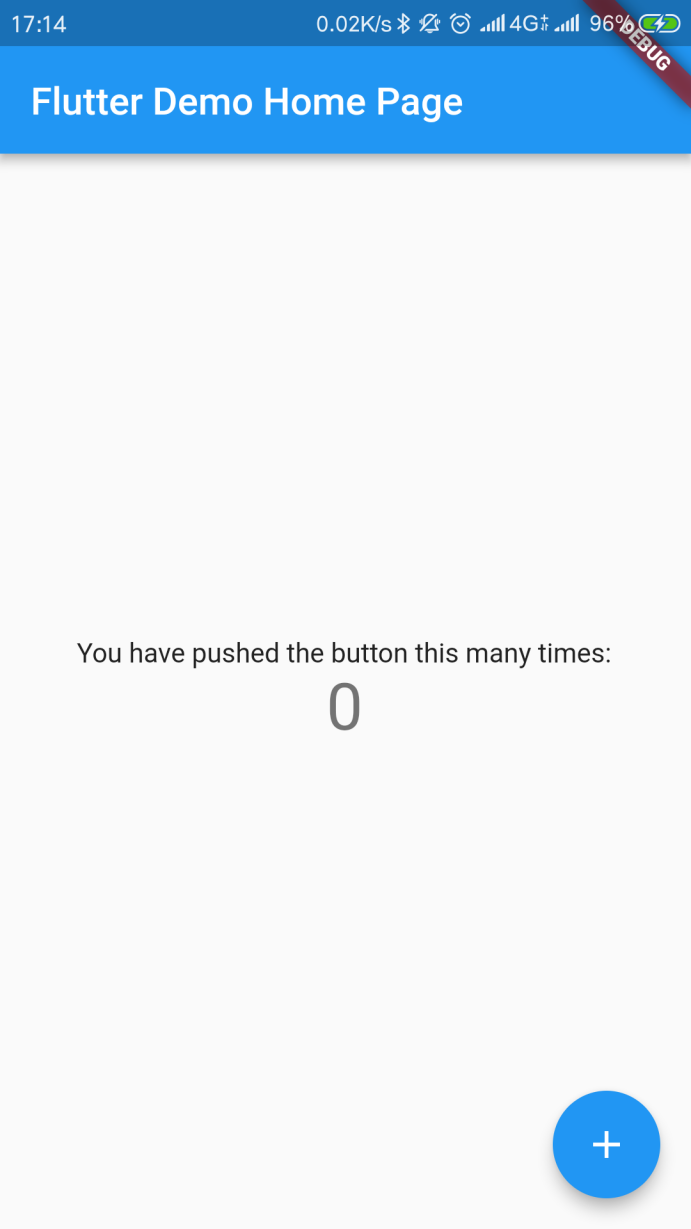
改成4.6再运行一次，又他妈出错了：



说是找不到gradle 4.6，还是上网找找答案吧。

看来就是与gradle的版本有关，我把build.gradle的版本调低一点，3.1.0，试试：

哦，shit，成功了：



# 2018.12.28 周五

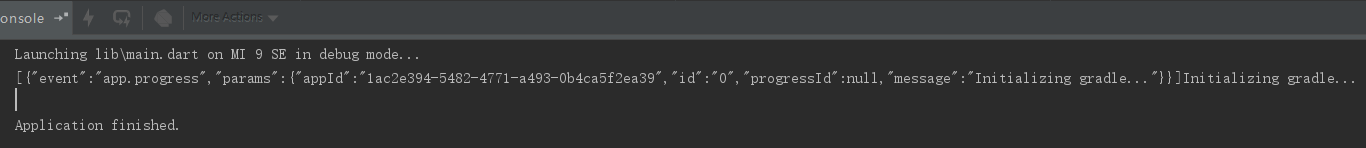
工欲善其事必先利其器，开发Flutter之前，你必须先学会Dart。目前国内还没有Dart2的汉化版在线教程或者书籍，所以最好的办法就是跟着官网去学习Dart2语法。

学完Dart语法后，我们就可以再回来继续学习Flutter。

# 2019.7.26 周五

时隔半年多，又重新学习Flutter了，这半年Flutter从无人知晓到现在到处都在炒作，Flutter的热度有了很大的改变。另外，Flutter也没有停止开发的脚步，版本更迭的非常快。今天我又重新装了一遍Flutter，基本没遇到困难。开始正式的学习和开发之旅吧！

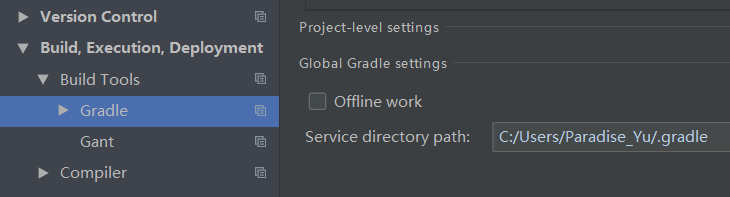
创建完工程之后什么都不干，直接运行应用，卡在了这个界面：



Initializing gradle...

很显然是Gradle的问题。

创建的Flutter工程依赖于Gradle，但是我们本地还没有这个东西，因此系统会下载一个对应版本的Gradle，这个过程巨慢啊。不如自己安装一个Gradle了，网上有相关的教程，核心思想就是找到AS安装gradle的目录，然后把我们自己下载的gradle放进去就可以了。我们需要参考工程的gradle-wrapper.properties文件，里面有版本号，还要找到下载安装的地方，可以通过File->setting->Build,Execution...->Build Tools->Gradle，找到Service directory path这个选项框，看看具体的路径是什么。



另外，安装的安卓手机必须打开USB调试模式，而且要允许USB安装。然后就运行成功了。

# 2019.7.30 周二

今天，把Dart的视频看完了，对Dart有了一个初步的、感性的认识。

在数据类型方面，Dart和JS非常像，可以使用var来定义变量，使用const来定义常量。[]来定义数组，{}来定义Map（JS中的对象）。

运算符和控制语句和其他语言是一样的，没什么可说的。

方法和JS也非常像，无非就是可选参数和默认参数值有点区别。也拥有匿名函数和闭包。

但是Dart有面向对象编程的部分，class、构造函数、静态、继承、接口实现、抽象类、混入、操作符重载。这些功能像是把很多语言的一些特性都吸纳了，例如Swift的?.来防止空指针引用；C++的class特性（静态、初始化列表、多重继承、操作符重载、泛型）；JS的call方法；Java中的implements等等。这应该才是Dart的精髓。

<https://www.imooc.com/learn/1035>

上面这个地址是学习视频，免费的。

接下来就是真正的进入到Flutter的学习了。

Flutter的学习视频：

<https://www.bilibili.com/video/av52490605/?p=15>

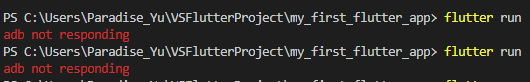
当学到在iOS上安装Flutter的时候，让我对Flutter的学习热情一下子降了下来，因为iOS安装包必须通过Xcode来打包。之前使用AppCan的时候，iOS的安装包直接在线打包，不用涉及到苹果电脑的问题。也就是说，尽管Flutter是跨平台的，但是开发并不是跨平台的，需要苹果电脑打iOS的包，说白了如果用苹果电脑开发的话才是一举两得，而Windows只能开发Android。

# 2019.7.31 周三

今天换了个视频看的：

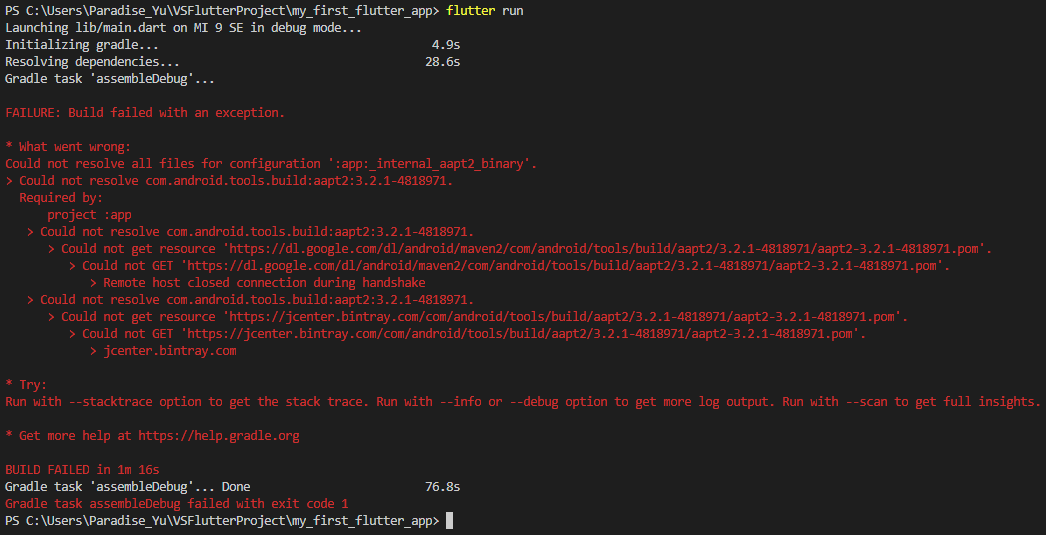
<https://www.imooc.com/video/18526>

这个视频用的是VSCode开发，所以今天主要是在VSCode中运行Flutter。但是我在运行Flutter run的时候报错了：



意思就是adb端口被占用了，解决办法网上也有。我没有那么做，而是不用模拟器，用手机连接，就解决了。

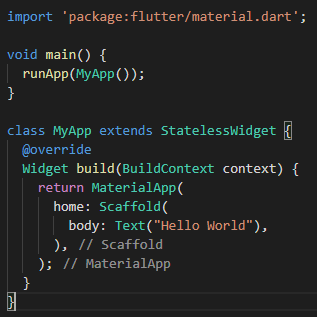
但是在运行之后还是报错了：



看这意思应该是要下载某些组件，但是由于当时我没有翻墙，所以失败了，所以说翻墙软件是开发必备啊！打开翻墙软件，再运行就好了。

# 2019.8.1 周四

今天首先写一个最基本的app，展示Hello World即可。重点是学习一些最基本的app骨架知识：



|  |
| --- |
| import 'package:flutter/material.dart';  void main() {  runApp(MyApp());  }  class MyApp extends StatelessWidget {  @override  Widget build(BuildContext context) {  return MaterialApp(  home: Scaffold(  body: Text("Hello World"),  ),  );  }  } |

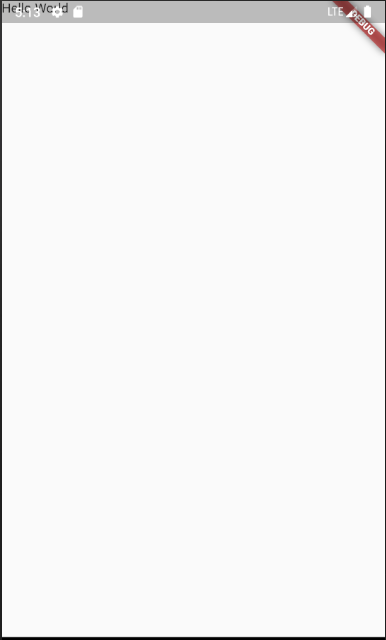
第一行的import表示引入一个文件，和C语言的include很像，引入的material.dart是一个基础的UI框架，类似于iOS中的UIKit。

第二行的main方法里面执行一个runApp方法，用来把app运行起来。里面的MyApp()是一个类的构造函数，也就是下面的这个class MyApp类。但是我不知道具体原理是什么，我如果把类的名字改成别的，运行就会出错，难道Flutter默认启动类的名字就是MyApp？这个名字太不正规了吧。

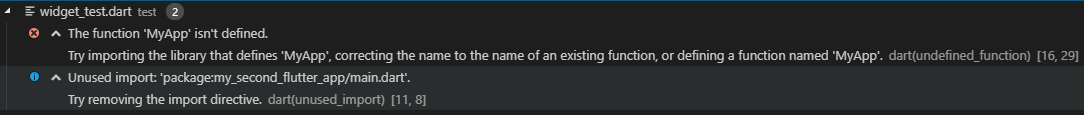
接下来就是MyApp的类定义，继承自无状态的Widget，里面的build方法是必须实现的，因为StatelessWidget是抽象的。build方法必须返回一个Widget，一般我们会返回一个MaterialApp对象。MaterialApp必须包含一些特定的属性，截一下官方文档的图：

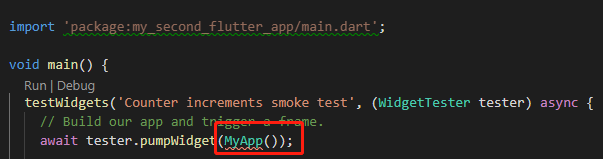


一般有一个home即可，home传一个Scaffold对象，这个是指定样式的。当然，也可以指定别的Widget，例如一个Text，只不过app不再是Scaffold样式了。在Scaffold的body中给定一些自定义的内容，即可显示一个Hello Word了。显示效果如下：



主类MyApp其实可以改成其他的名字，但是如果你只是改了main.dart中的类名的话，test文件夹下的widget\_test.dart文件就会报错：





我们只要把这个MyApp的名字同步改成我们修改后的类名就好了。