

使用 Android Studio 阅读 AOSP 源码

发表于 2019-06-23 更新于 2019-12-11 分类于 [Android 9](#)
[系统源代码情景分析](#) 阅读次数: 986 Disqus: [0 Comments](#)

[上一篇文章](#)介绍了 AOSP 9 在 macOS 10.14 环境下的编译方式，本文就 Andorid Studio 环境下阅读 AOSP 源码的方式进行记录。

阅读前 AOSP 的准备

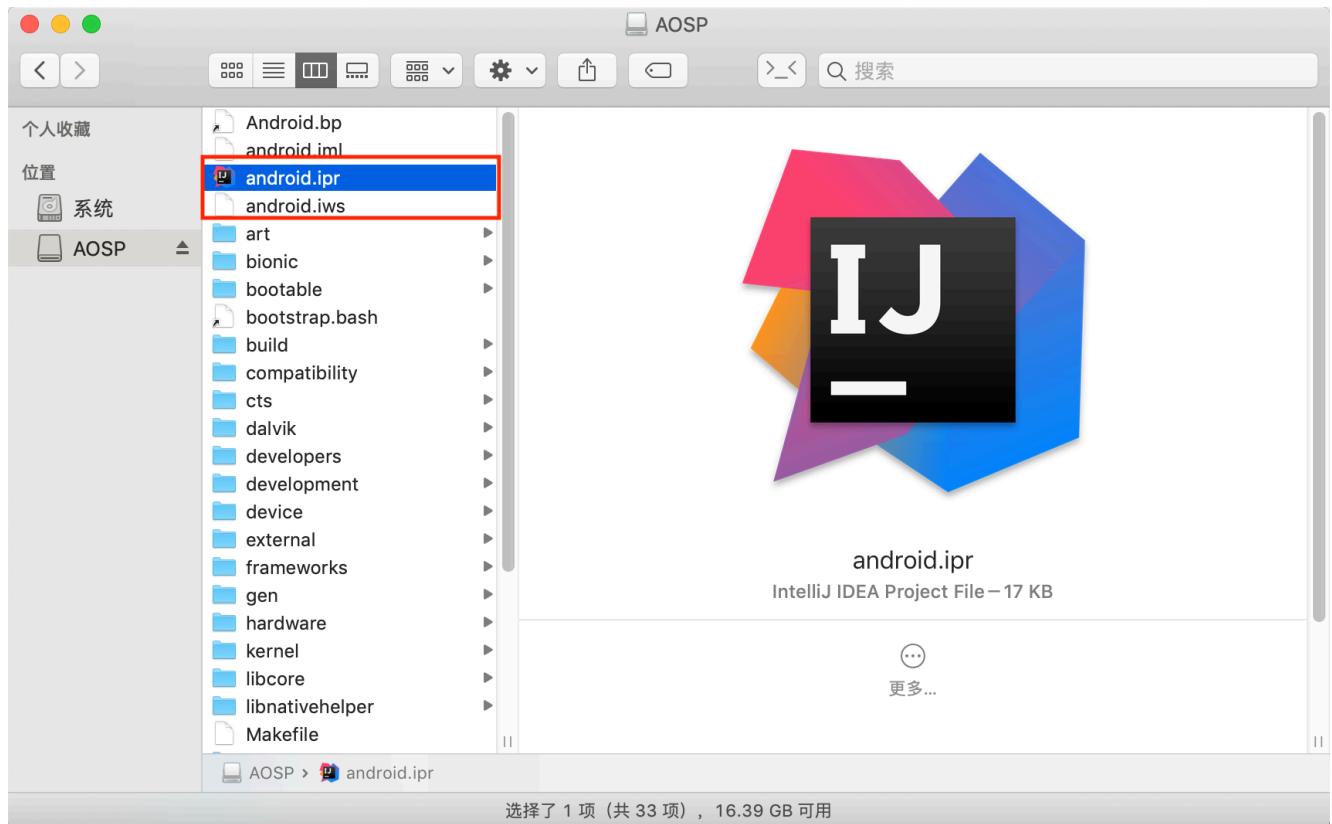
开始之前我们先为 AOSP 生成 IDEA 工程文件，便于稍后在 Android Studio 中载入。

1. 为 AOSP 生成 Android Studio 工程配置文件

```
1 cd /Volumes/AOSP
2 # 设置 AOSP 编译所需的环境变量
3 source build/envsetup.sh
4 # 使用 idegen.sh 脚本生成 IDEA 工程文件
5 development/tools/idegen/idegen.sh
6 -----
7 Read excludes: 4ms
8 Traversed tree: 68826ms
```

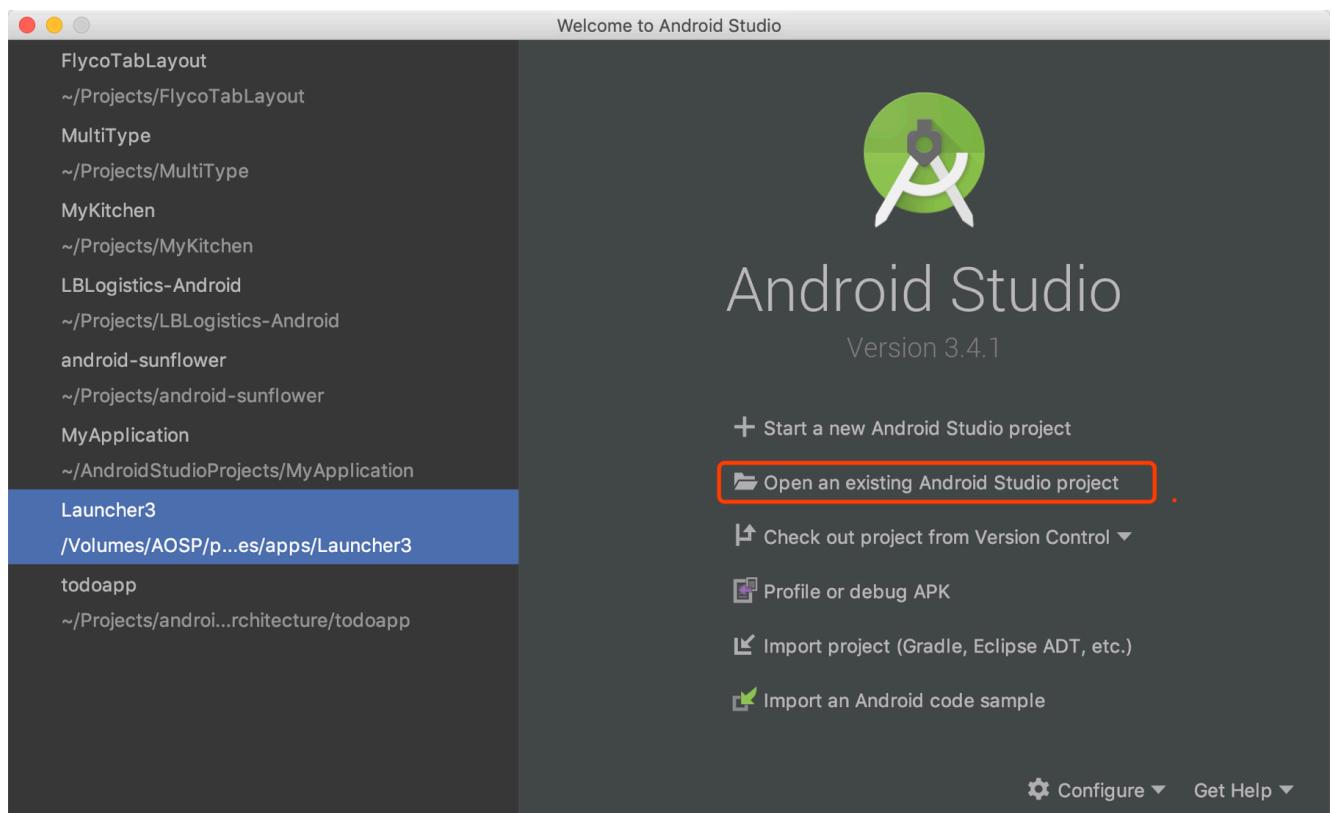
稍等片刻，即可在 AOSP 目录下看到生成的 `android.iml` 和 `android.ipr`，其中 `iml` 文件表示 `information of modules`，用来描述 AOSP 的模块信息。`ipr` 文件表示 `IDEA project configuration`，用来描述 `IDEA` 的工程配置信息，双击此文

件时系统将直接使用 Andorid Studio 打开此项目。

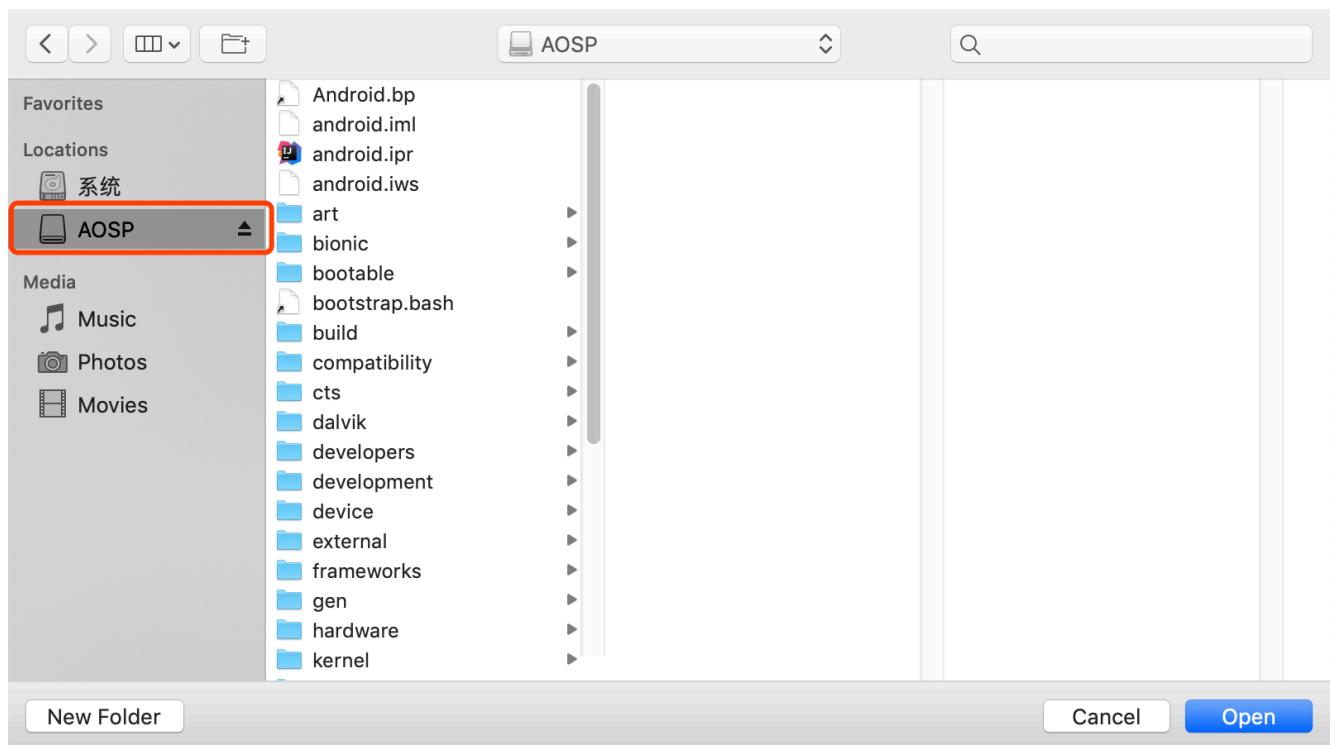


2. 导入 Android Studio

1. 点击 Open an existring Android Studio project



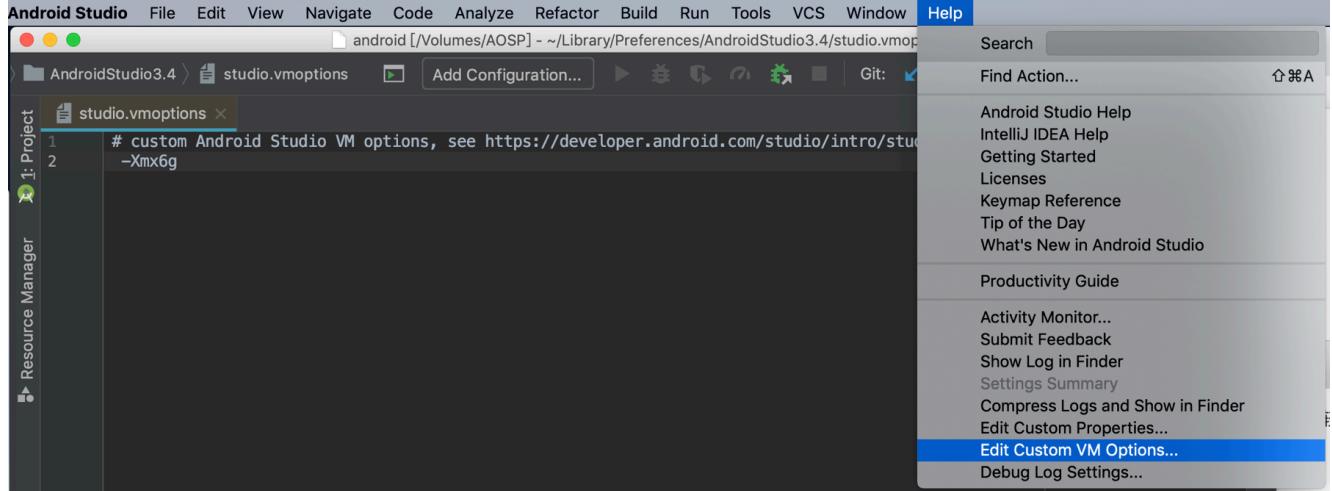
1. 选中 AOSP 根目录



使用 Android Studio 打开 AOSP 项目时会先对项目源码建立索引，此步骤比较耗时，笔者所用的 4 核心 8 线程的 CPU 建立索引大概需要 8-10 分钟，读者可趁此机会起身活动片刻。

3. 修改 Android Studio VM 参数，优化源码阅读体验

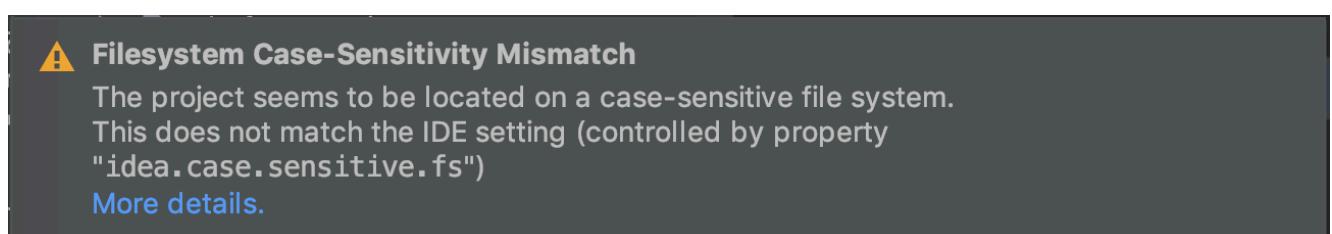
AOSP 源码文件较多，Android Studio 默认设置的 1280m 堆最大内存对于 AOSP 这样的大型项目已不再合适，为了更好的性能，我们可以通过如下方式来增大 Android Studio 的堆最大内存。



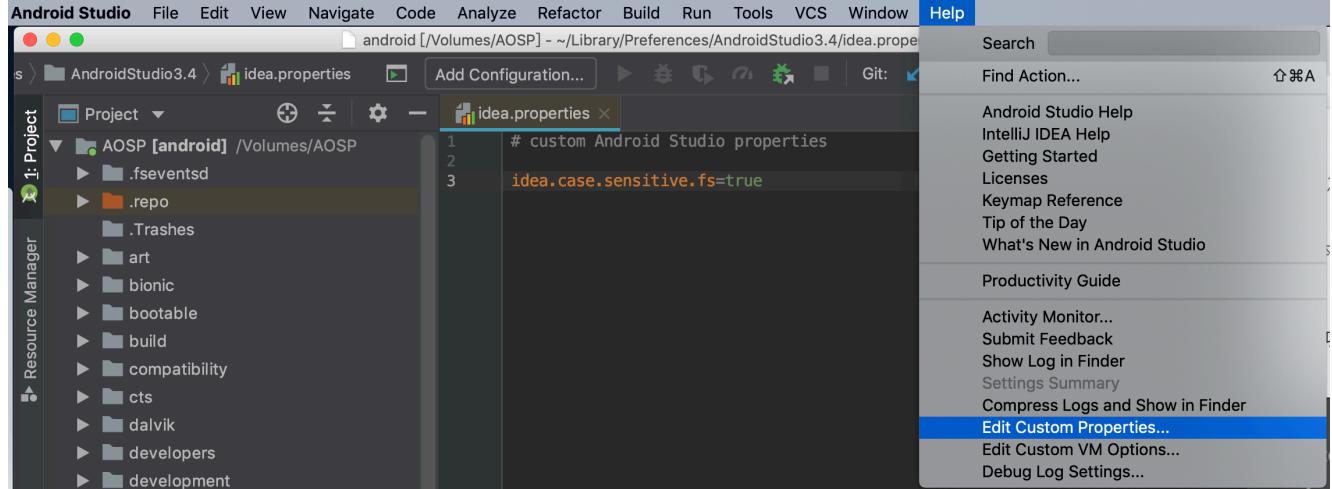
1. 点击 Help > Edit Custom VM Options 以打开您的 `studio.vmoptions` 文件。
2. 向 `studio.vmoptions` 文件添加一个行，使用语法 `-XmxHeapSize` 设置最大堆内存。如图所示，笔者将 Android Studio 的堆最大内存空间设置为 6G。
3. 保存对 `studio.vmoptions` 文件所做的更改，然后重新启动 Android Studio 以使更改生效。

4. 修改 Android Studio 文件系统大小写敏感项

AOSP 首次导入 Android Studio 后，会出现下图所示的气泡弹窗，表示 AOSP 项目的文件系统大小写敏感度与 Android Studio 默认设置的文件系统大小写敏感度不匹配。



为了更好的阅读体验，我们可以采用如下方式修改 Android Studio 的文件系统大小写敏感度。



1. 点击 Help > Edit Custom Properties。如果您之前从未编辑过 IDE 属性，Android Studio 将提示您新建一个 `idea.properties` 文件。点击 Yes 创建文件。
2. 此时 `idea.properties` 文件将在 Android Studio 的编辑器窗口中打开。
3. 编辑 `idea.properties` 文件，添加自定义属性 `idea.case.sensitive.fs=true`。
4. 保存对 `idea.properties` 文件所做的更改，然后重新启动 Android Studio 以使更改生效。

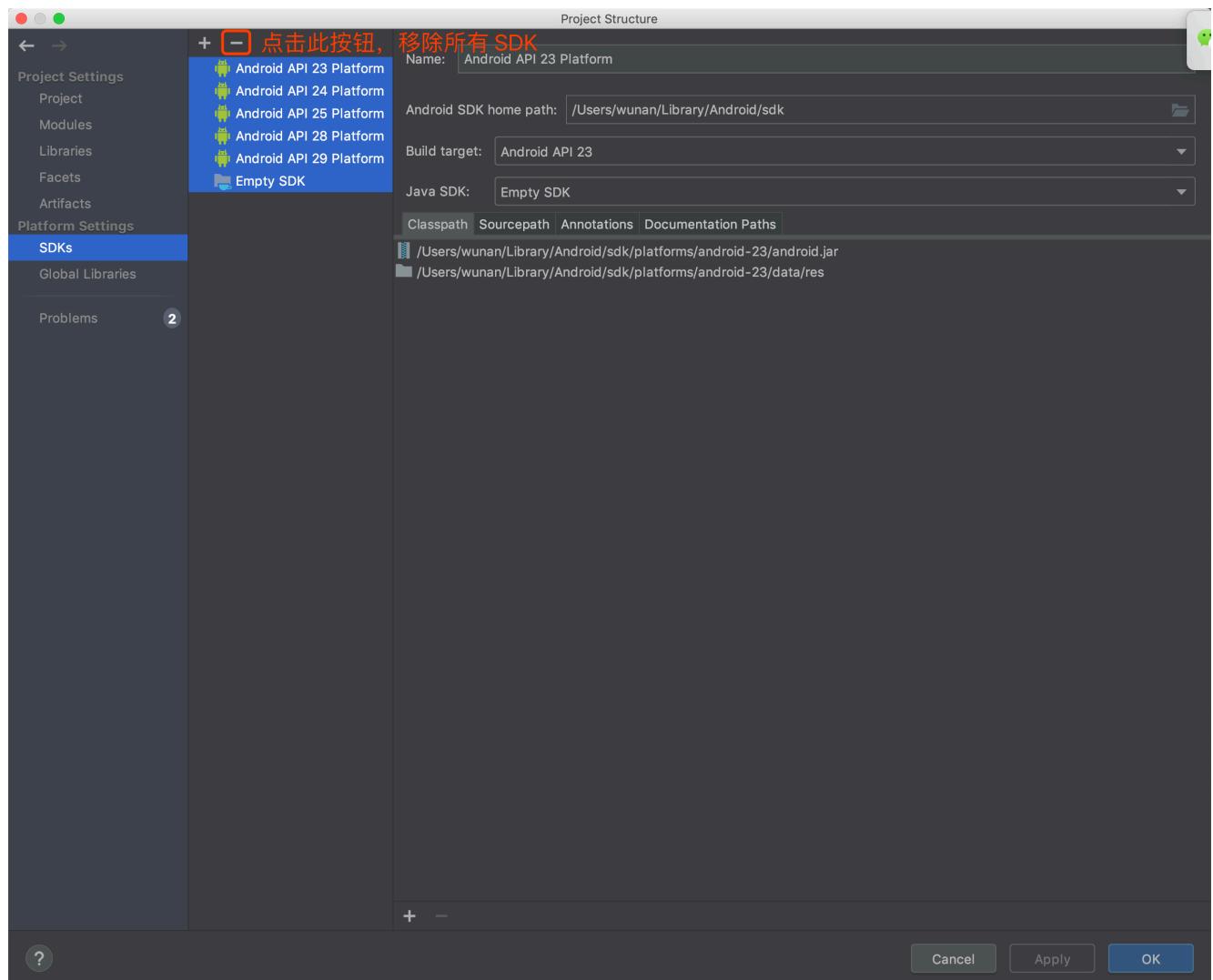
5. 修改 AOSP 工程配置

1. 清空 Android Studio 中关联的 JDK 与 SDK

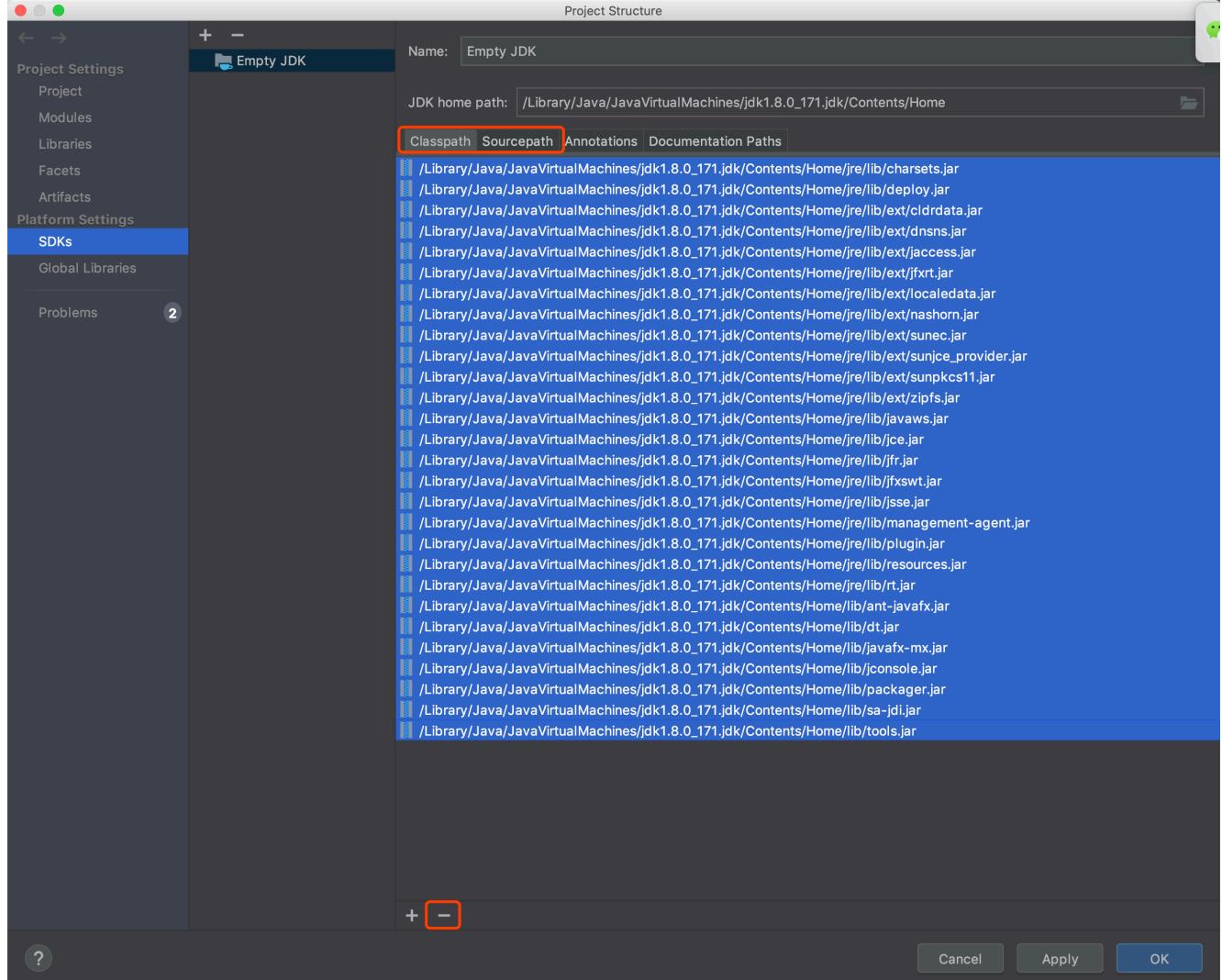
我们首次安装 Android Studio 时，通常会为其配置默认的 JDK 和 SDK，而 AOSP 项目中也包含 Android SDK 与 Java JDK 的源码，但在未清空 Android Studio 的默认 JDK 与 SDK 之前，Android Studio 会优先加载其默认的 JDK 与 SDK 而非 AOSP 中包含的，我们可以采用如下方式清空默认的 JDK 与 SDK。

1. 点击 File > Project Structure..，也可使用快捷键 $\text{⌘} + ;$

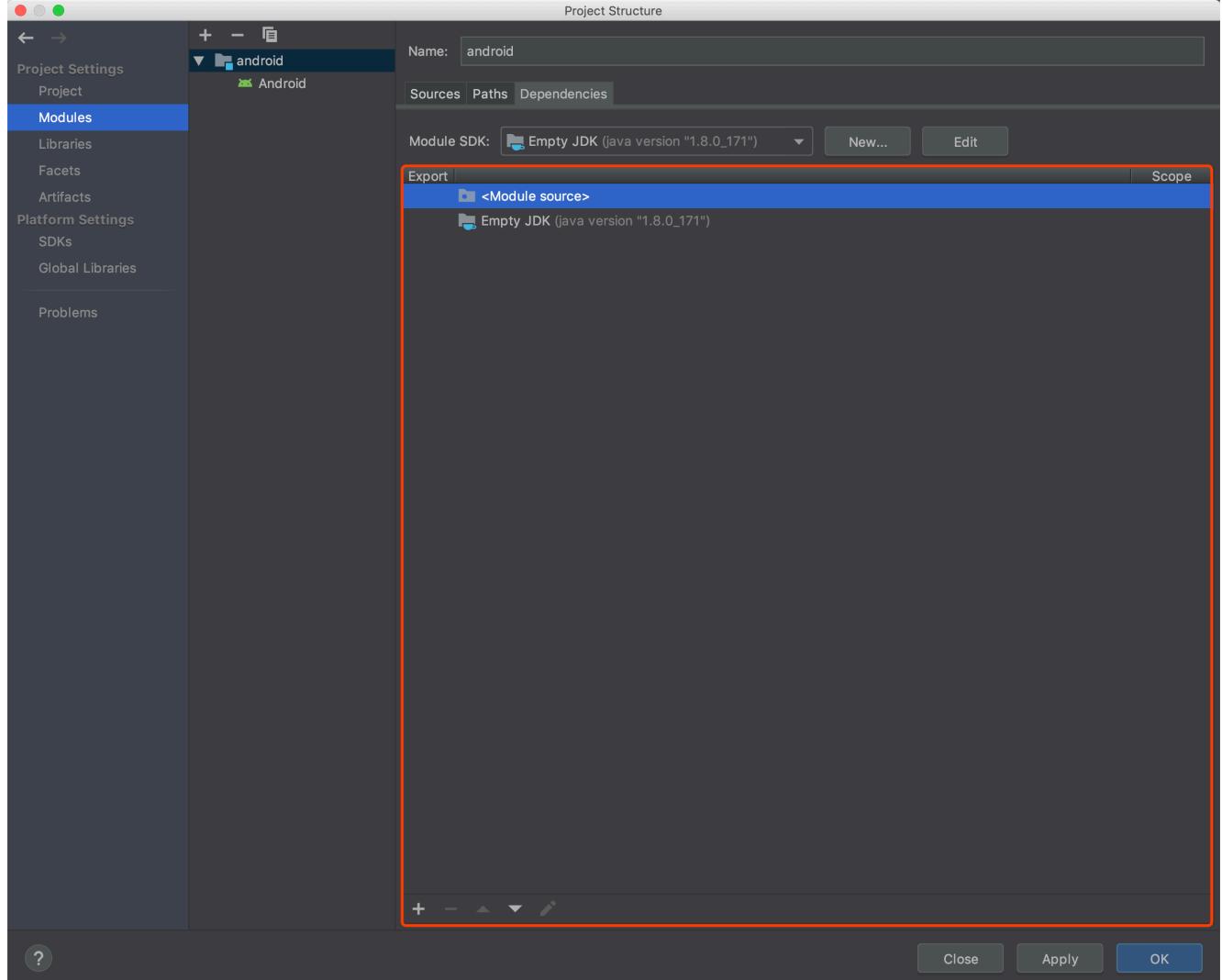
2. 选中 Platform Setting > SDK，然后选中所有的 SDK，最后点击上方的移除按钮 —



1. 点击上方的添加按钮 +，添加 JDK
2. 将新增的 JDK 更名为 Empty JDK
3. 使用下方的移除按钮 —，依次清空 Empty JDK 下方标签栏中 Classpath 与 Sourcepath 下的所有依赖项



1. 点击右下方的 Apply 按钮保存更改，然后进行下一步操作。
2. 移除 AOSP 内的多余依赖项
 1. 点击 Project Settings > Modules，选中 android 模块，将其 Module SDK 设置为我们上一步创建的 Empty JDK
 2. 移除 Export 中除 Module source 、 Empty JDK 以外的所有条目，如下图所示。



1. 点击右下方的 OK 按钮保存修改。

小结

经过以上操作，代码中的红色错误全部消失，代码间也能够进行任意跳转（除 native 代码外），总算可以舒畅的阅读 AOSP 了。

常见问题

1. `import` 部分出现大量的红色错误，代码内无法跳转，提示 `Cannot find declaration to go to`。参考[上一篇文章](#)，重新编译 AOSP 解决。
2. 项目索引时间太长，或代码跳转太慢。AOSP 项目模块非常多，文件数量也非常大，Android Studio 每次打

开 AOSP 时都需要为 AOSP 项目内的海量文件建立索引，跳转时也需要在海量文件中进行检索，慢是必然的。为此，我们可以修改 AOSP 的工程模块描述文件 android.iml，排除一些不关心的模块，以此来提高索引建立与检索时速度。

AOSP 的工程模块描述文件 /volumes/AOSP/android.iml 内容如下。

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <module relativePaths="true" type="JAVA_MODULE" version="4">
3  .....
4      <component name="NewModuleRootManager" inherit-compiler-output="true"
5          <exclude-output />
6          <content url="file://$MODULE_DIR$">
7          .....
8          +     <sourceFolder url="file://$MODULE_DIR$/frameworks/base/core/java"
9          +     <sourceFolder url="file://$MODULE_DIR$/frameworks/base/core/test"
10         -     <excludeFolder url="file://$MODULE_DIR$/..repo" />
11         .....
12         </content>
13     </component>
14 </module>
```

其中，绿色的 sourceFolder 表示已导入的模块，红色的 excludeFolder 表示需要从项目中排除的模块。我们可仅保留部分感兴趣的模块，将其他不感兴趣的模块的 xml 标签由 sourceFolder 修改为 excludeFolder，这样一来项目索引和检索的速度便会快很多。