

✨HustHoleAndroid 新人入门文档✨



欢迎加入HustHole Android开发者团队！

如果你是已经写了许多小demo的好手，但是因为第一次碰到 HustHoleAndroid 这样一个中小型项目而无从下口，产生挫败感.....那这篇文档就是为你量身定做的。

通过本文你可以了解到：

- **HustHoleAndroid** 项目环境搭建流程
- 了解 **HustHoleAndroid** 项目工程的架构设计
- 1037树洞需求研发流程
- 其他有趣的小知识

1. 项目搭建

HustHoleAndroid项目仓库代码的拉取

开头第一步，获取到项目源码并成功在开发环境下运行起来！

1.1 环境搭建

- **Android Studio**：Android开发官方唯一指定IDE，简称AS
- **Android SDK**：Android开发工具包（集成在AS中）

！ 请在安装过程中保持网络畅通，尽量全程打开vpn，**一定不要使用华中科技大学校园网！！**
一定不要使用华中科技大学校园网！！ 一定不要使用华中科技大学校园网！！ 优先使用手机热点或者家庭WIFI。

遇到任何卡网络的问题，请和踩过所有坑的mhh联系解决😁

1. 下载Android Studio

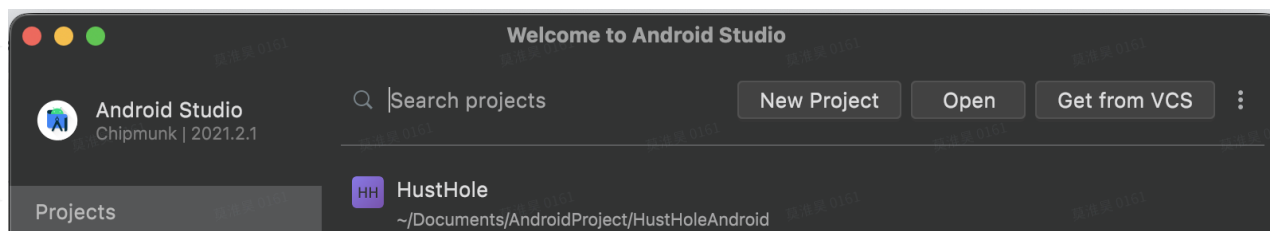
2. 使用 SDK Manager 安装 SDK/NDK 等组件：

- a. 在打开的项目中，从主菜单选择 **AndroidStudio > Preferences > Tools > 搜索Android SDK Manager** 或者点击状态栏Sdk Manager
 - b. 点击 **SDK Tools** 标签。
 - c. 勾选 **CMake（必须）、和 NDK（可选）** 旁的复选框（新版NDK已集成LLDB，无需额外安装）
 - d. 从 SDK 管理器中选择CMake 和 NDK。
 - e. 点击 **Apply**，然后在下一个对话框中点击 **OK**。
 - f. 安装完成后，点击 **Finish**，然后点击 **OK**。
3. 安装插件（极大提高开发效率）
- a. **CodeLocator**：国人开发的应用页面抓取插件，2min学会，新人看项目再也不会找不到入手点
 - b. **Translation**：AS内置翻译插件，支持一键翻译Javadoc，注释等，方便好用，英文菜鸡福音
 - c. **Rainbow Brackets**：彩虹括号，多层括号嵌套一目了然

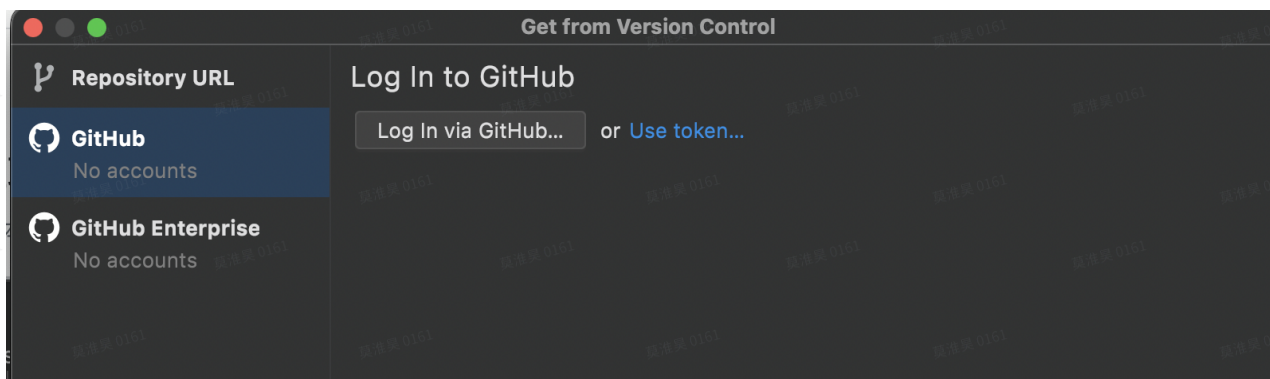
。 。 。

1.2 使用Git拉取项目仓库

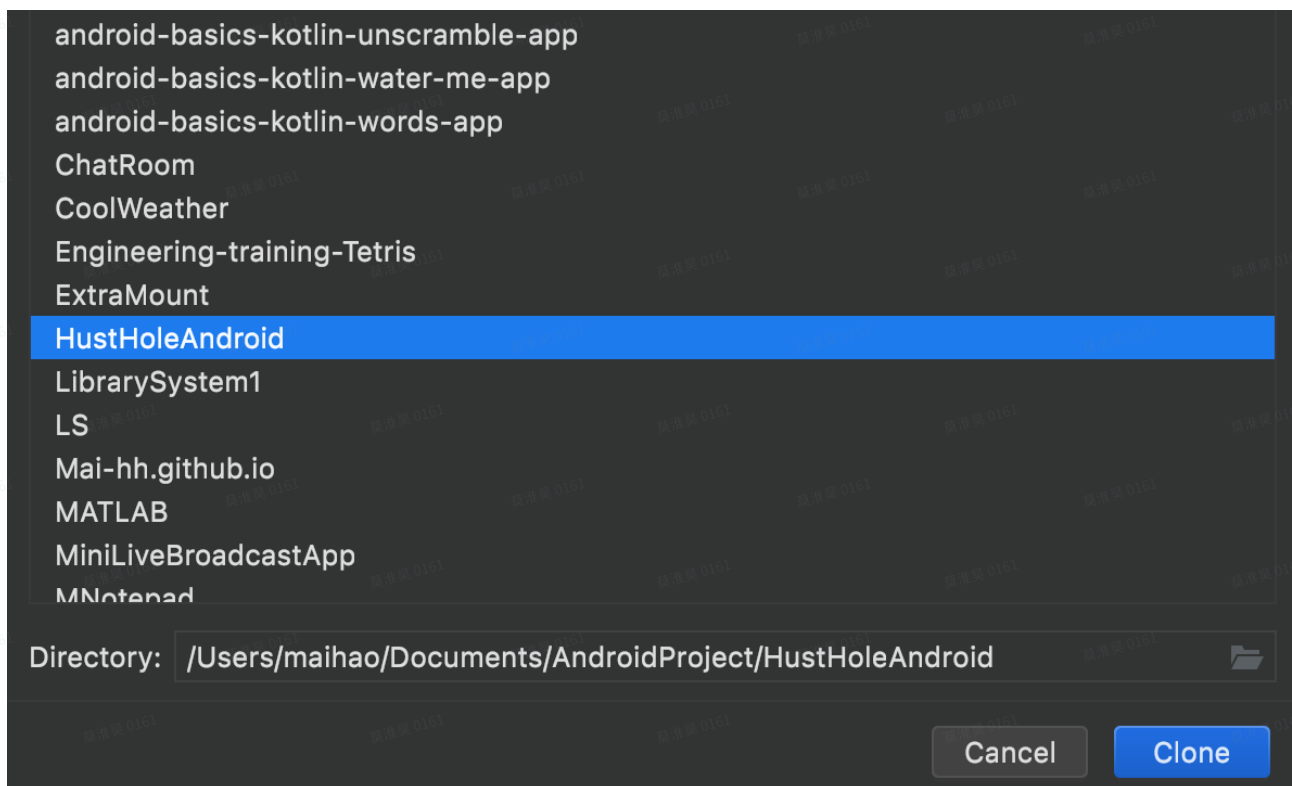
命令行是Git的根本，这里使用最简单的AS内置的Git图形化工具，但请**不要忽视Git基础的学习**



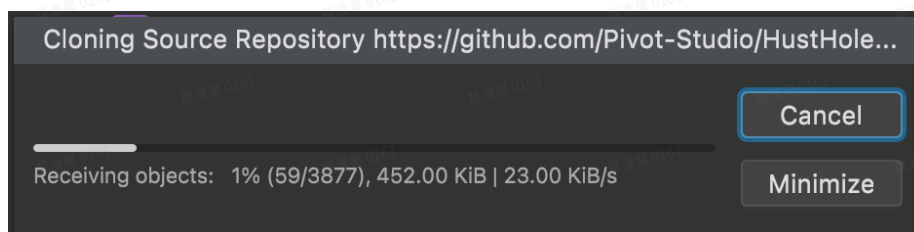
1. 点击 Get from VCS



2. 登陆Github账号，或者使用 Token 登陆（需要登陆Github网站，在profile设置下的security项中新建Token，具体细节可自行搜索配置步骤easy）



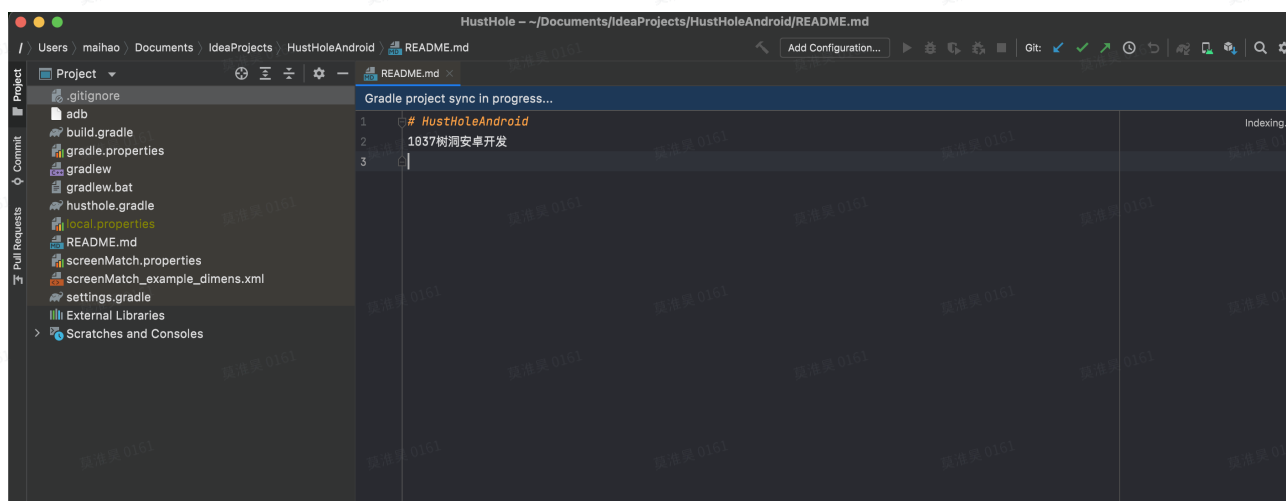
- 选择仓库HustHoleAndroid，点击右下角clone（如果没有这个仓库或者clone失败，请联系其他开发者确认你是否拥有权限）



出现这个框说明clone成功了，其他情况下都是失败的。clone失败不要着急，因为github是服务器不在国内，任何问题都有可能出现，请及时联系其他开发者帮助排查😄

1.3 项目构建

- 上面的步骤全部完成后，会直接进入项目工作区，并且进入Gradle project sync状态

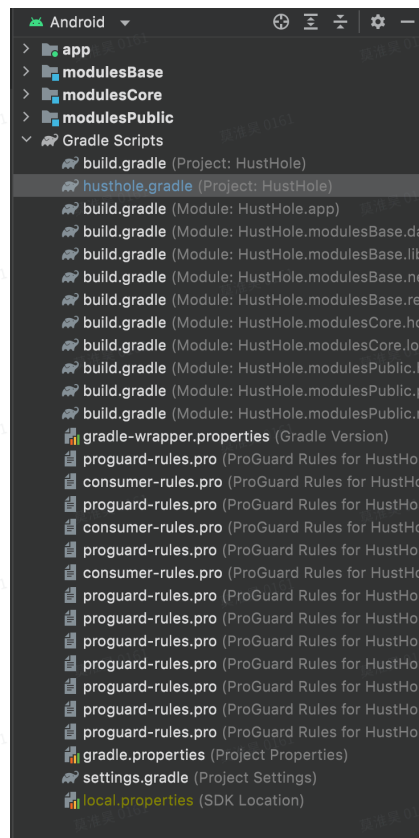


虽然我很想详细的和朋友们在这里解释 Gradle 以及 Android 应用的打包流程，难道作为一名开发者，你不想了解工作区中一堆奇奇怪怪的文件和业务代码如何“变”成手机上的App吗？

可惜这个内容细究下去太过复杂，也不符合新人文档的特性，所以就打住（这是成为安卓架构师必须要掌握的部分）

简单来说，目前AS自动的从本地 或者 云端库中同步（sync）项目所配置的依赖库。没错，只要涉及到“云端”，请保证vpn开启，并且不在校园网状态下（这不会耗费很多流量）

- 找到husthole.gradle文件



- 注意两个变量 isRelease & isMac

```
ext {
    isRelease = true
    isMac = true

    androidID = [
        compileSdkVersion: 31,
        minSdkVersion      : 29,
        targetSdkVersion   : 31,
        versionCode         : 2, //不能是非整数，无语子
        versionName         : "v1.1.9-测试版"
    ]

    networkID = [
        //okhttp网络请求
        "okhttp-bom"       : "com.squareup.okhttp3:okhttp-bom:4.9.3", // de
        "okhttp"           : "com.squareup.okhttp3:okhttp", // define any r
        "okhttp-logging-interceptor" : "com.squareup.okhttp3:logging-interceptor"
    ]
}
```

相信我，使用Mac进行开发相比于Win会获得更好的体验（Google love ntr :)）

正常使用windows环境情况下

```
1 isRelease = true
2 isMac = false // Mac 下为 true
```

涉及到 Gradle 文件的改动都需要重新同步（Sync Now）一下，原因和上面提到的打包流程相关。

- 运行App

激动人心的时刻



💡 所以 isRelease 这个变量的作用是什么？这就涉及到一个App开发中一个比较**黑科技**的部分--组件化。当 isRelease = false 时，AS进入开发者前期配置好的组件化调试状态。

当理解**HustHoleAndroid**的项目架构之后，组件化就不再神秘

1.4 网络接口测试工具 ApiFox

在IDE通过打印日志输出测试接口通信能力实在是有些落后，我们当然希望能够在保证接口通信功能正常，返回内容明确的条件下进行网络相关模块的开发。所以我们要引入测试工具，常见的有postman和apifox，这里选择国产的后者

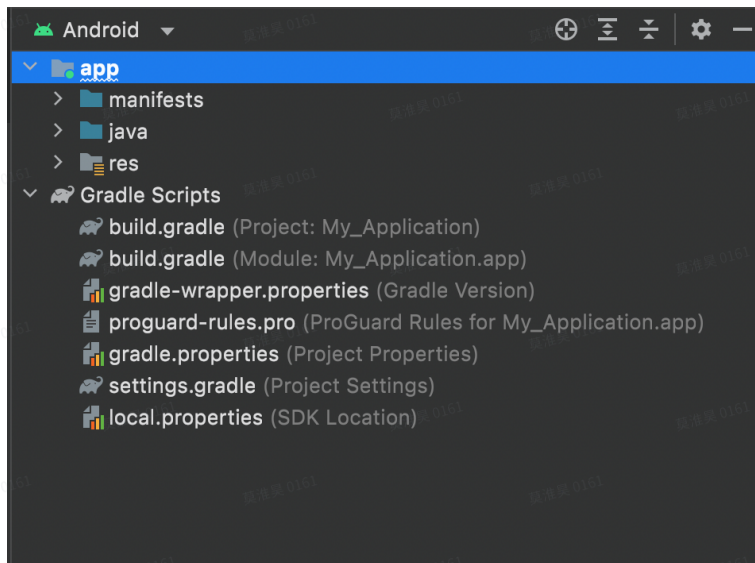
- 官方网址<https://www.apifox.cn/>

请联系管理员加入“1037树洞”接口文档，使用已经创建好的接口测试集

2. 项目结构

2.1 理解App工程的结构

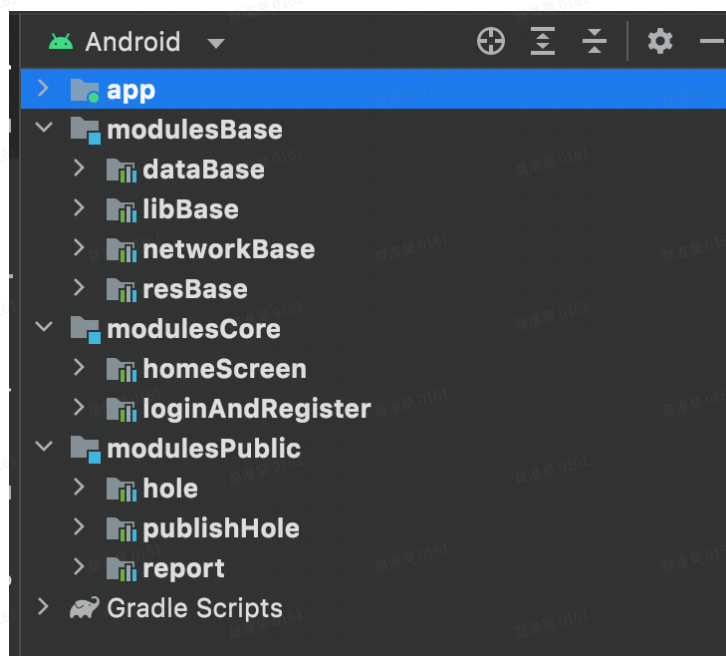
我们最熟悉的项目文件结构一般是这样的



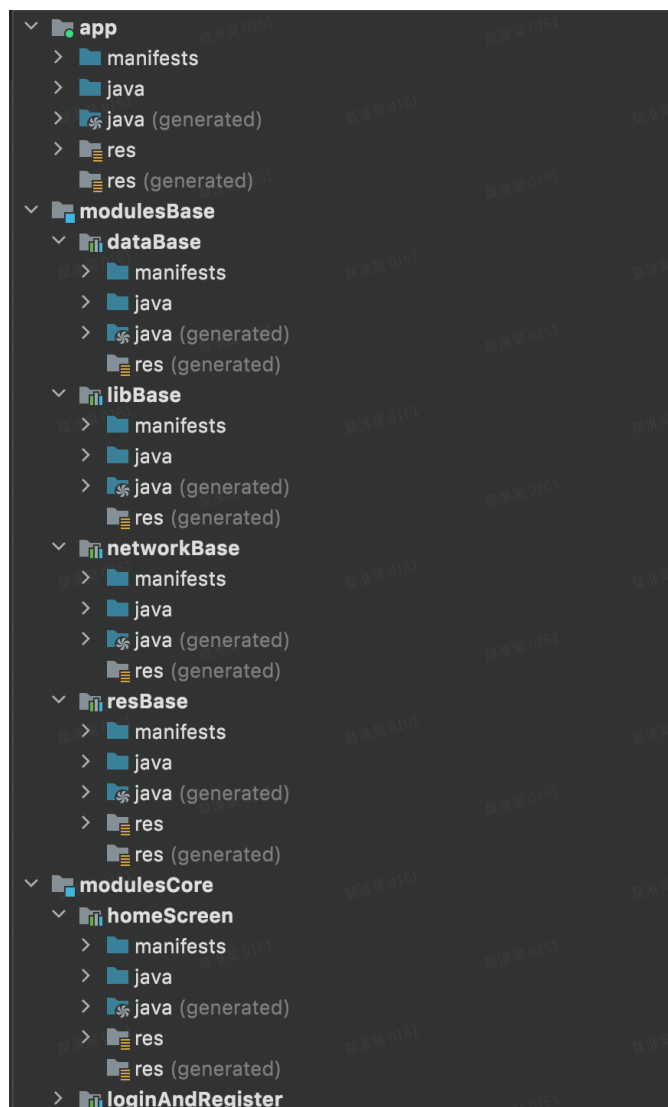
简单解释一下：

- App 文件夹：包括项目的源代码（java）、资源（res）、配置清单（manifests）
- Gradle Scripts：配置脚本，抽象的说就是**App打包的指导者**，这里面的文件完全是为了配置App服务，例如：依赖库引入、依赖库版本更新。比如说项目中使用的Activity类就是一个依赖库提供的，你可以在某个.gradle文件中修改引入的版本号，在不改变源码的情况下获取新的功能或者API。然而在这样一个普通的App下，并不能体现出**App打包的指导者**的作用

树洞的项目文件结构是这样的



展开来是这样的



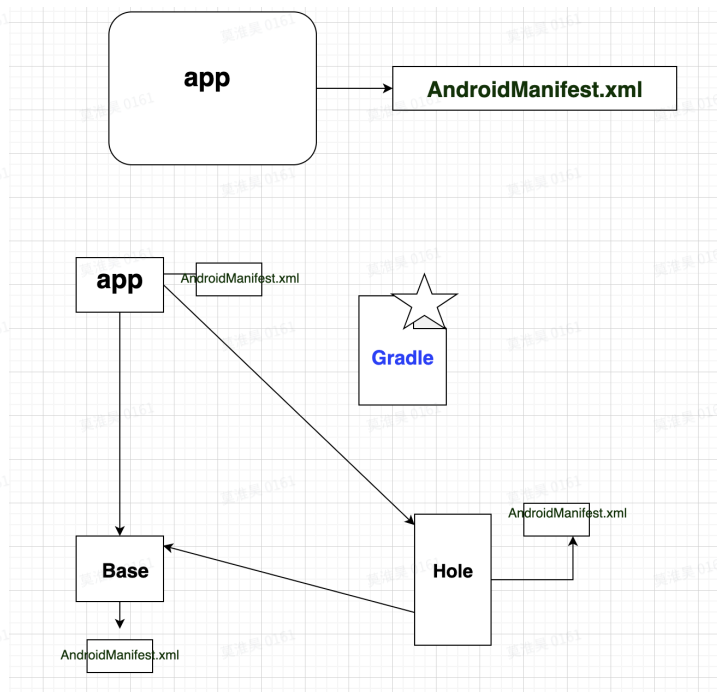
如果你已经开始慌了，请千万别着急，注意观察文件的层级关系和每一个module下文件的结构。你会发现，他们的结构和普通的app都是一模一样的！！假设我称dataBase这个文件为一个组件，那么每一个组件都有作为一个完整app的潜力。

那他们是如何相互配合，最后成为一个App的呢？

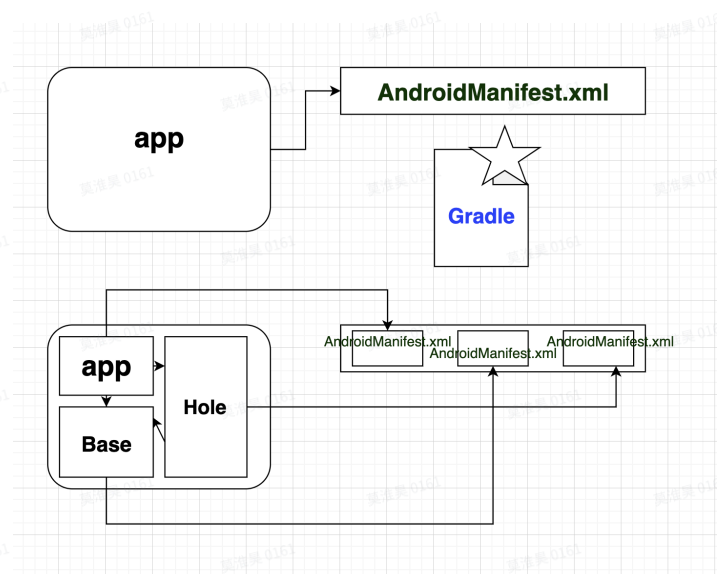
还记得上面提到的Gradle吗？**App打包的指导者**，就是通过修改gradle文件，在app构建（build）的时候，将每一个模块的代码合并成一个app。**合并的依据**就是那个你很熟悉的文件——AndroidManifest.xml。

简单来说，Gradle通过查找每一个组件的AndroidManifest.xml，自动合并成一个AndroidManifest.xml

为了便于理解，我还准备了下面两张图



开发者眼中的文件结构



gradle眼中的文件结构

此时此刻，希望你心中对树洞的项目结构的恐惧少了许多 😊

如果你理解了组件化的原理，请看[当前树洞的架构](#)

App层

外壳工程，
存放整个app
的入口活动

保证组件分层的结构，任何下层组件不得依赖上层组件，也不能持有上层对象的引用，同层之间理论上允许相互引用，但是要尽量保持模块间解耦合

app

业务层

存放
HustHoleAndroid
所有的业务逻辑，
页面逻辑的地方

modulesCore

homeScreen

loginAnd
Register

modulesPublic

hole

publishHole

report

基础层

存放所有的通用组件，抽象类，工具类，图片&字符串等资源的地方

modulesBase

dataBase

networkBase

libBase

resBase

- App: 壳工程，也是最顶级的工程，目前存放了一个StartActivity负责指引App启动
- modulesCore: 存放HustHole**独一份的页面**，例如底部导航栏管理的4个页面，以及小树林模块的几个界面（homeScreen），登陆注册页面（loginAndRegister）
- modulesPublic: 存放HustHole布局相同，但是内容可能不同的页面，比如树洞详情页（hole），树洞发布页（publishHole），举报页面（report）

这么说小树林详情页也应该放在modulesPublic，然而截至2022.7.26 小树林详情页放在homeScreen中

- modulesBase: 和页面完全无关的处理数据的工程，包括数据库（dataBase），网络请求框架（networkBase），基础依赖库（libBase），存放设计图字符串等的资源库（resBase）

💡 那么问题来了，假设我现在在homeScreen组件中写代码，我想用到networkBase中的类怎么办？

- 注意图中的那一大段话，上层模块是可以随意拿到下层模块的引用的！所以，直接写就完事了
- 但是，在下层模块就无法使用到上层模块的类了（如果在IDE的自动补全提示中发现可以使用，也请不要使用(说明项目的访问性限制做的不够好)，思考一下你的代码是不是写错地方了）

2.2 1037树洞使用到的技术

- 架构模式：MVVM，使用databinding+livedata库
- 网络请求：Retrofit+okhttp建立与后端的通信接口，Rxjava处理异步消息（用的很简单）
- 上下滑界面刷新：smartrefreshlayout开源库
- 屏幕适配：screenmatch处理屏幕适配
- 图片加载：Glide处理绝大多数网络图片加载
- 页面跳转管理：Navigation库管理主界面四个Fragment

3. 学习资料

资料	备注
《第一行代码 第三版》	第三版的编程语言是Kotlin，第二版是Java(内容严重过时)
Kotlin官方最佳实践	最简单直观的Kotlin教程(需要科学上网)
Android基础教程	最简单直观的Android Kotlin教程(需要科学上网)

！ 1037树洞作为当前（2022/7/18）华中科技大学现存最大的匿名交流社区，作为一名维护者，肩负的责任越重，对能力的要求越高，希望你能在软件维护和性需求的创新中，不断提高作为一名软件工程师的素养