

在 VSCODE 中调试

第一步：

- ① 切换到 vscode 的调试和运行模式,
- ② 配置好 .vscode 中的启动项 launch.json
- ③ 共有四种模式

```
[1] debug : 模式编译产物适合纯 Flutter 侧代码的开发、调试
[2] profile : 的用来做性能分析和测试
[3] release : 的用于打包发布
[4] test
```

```
[1] debug : 模拟器/真机运行
[2] profile : 只能真机运行
[3] release : 只能真机运行
[4] test
```

操作详情参见：[flutter vscode launch.json 调试](#)

四种模式区别参见：[Flutter 的四种运行模式：Debug、Release、Profile 和 test](#)

Debug

Debug 模式可以在真机和**模拟器**上同时运行：会打开所有的断言，包括 debugging 信息、debugger aids（比如 observatory）和服务扩展。优化了快速 develop/run 循环，但是没有优化执行速度、二进制大小和部署。命令 flutter run 就是以这种模式运行的，通过 sky/tools/gn --android 或

者 sky/tools/gn --ios 来 build。有时候也被叫做“checked 模式”或者“slow 模式”。

Release

Release 模式只能在真机上运行，不能在模拟器上运行：会关闭所有断言和 debugging 信息，关闭所有 debugger 工具。优化了快速启动、快速执行和减小包体积。禁用所有的 debugging aids 和服务扩展。这个模式是为了部署给最终的用户使用。命令 flutter run --release 就是以这种模式运行的，通过 sky/tools/gn --android --runtime-mode=release 或者 sky/tools/gn --ios --runtime-mode=release 来 build。

Profile

Profile 模式只能在真机上运行，不能在模拟器上运行：基本和 Release 模式一致，除了启用了服务扩展和 tracing，以及一些为了最低限度支持 tracing 运行的东西（比如可以连接 observatory 到进程）。命令 flutter run --profile 就是以这种模式运行的，通过 sky/tools/gn --android --runtime-mode=profile 或者 sky/tools/gn --ios --runtime-mode=profile 来 build。因为模拟器不能代表真实场景，所以不能在模拟器上运行。

test

headless test 模式只能在桌面上运行：基本和 Debug 模式一致，除了是 headless 的而且你能在桌面运行。命令 flutter test 就是以这种模式运行的，通过 sky/tools/gn 来

build。

在我们实际开发中，应该用到上面所说的四种模式又各自分为两种：一种是未优化的模式，供开发人员调试使用；一种是优化过的模式，供最终的开发人员使用。默认情况下是未优化模式，如果要开启优化模式，build的时候在命令行后面添加-unoptimized参数。

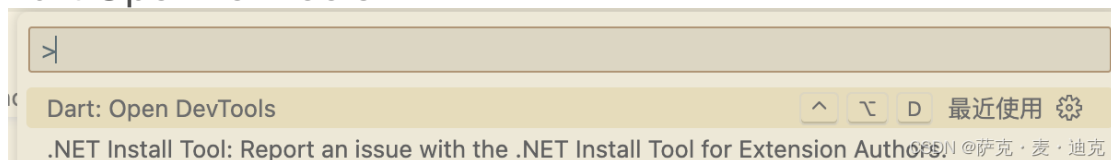
第二步：

调试模式开始运行后

command + shift + p

点击

Dart OpenDevTools



选择工具

