

基于GetX架构和健全的空安全的 Flutter模板项目

目 录

CONTENTS

- 1 Flutter 官方与开源社区新动态
- 2 融创数科对 Flutter 的技术积累与使用现状
- 3 Dart 基本语法与 Flutter 开发实用小技巧

第一章

PART 1

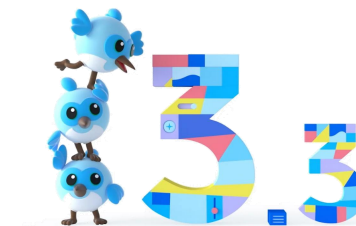
Flutter 官方与开源社区新动态



大家好，我是 Zoey

See What's new in Flutter 3.3 and what's new in the docs
The Flutter and Dart teams are hiring. [Learn more](#)

Flutter 3.3 正式发布，快来看看有什么新功能吧



Dart 2.18 发布，Objective-C 和 Swift interop

```
environment:  
  sdk: ">=2.16.0 <3.0.0"
```

```
neday — neday@bogon — ~ — -zsh — 80x8  
Last login: Wed Sep 21 17:06:10 on ttys008  
[→ ~ flutter --version  
Flutter 3.3.2 • channel stable • https://github.com/flutter/flutter.git  
Framework • revision e3c29ec00c (8 days ago) • 2022-09-14 08:46:55 -0500  
Engine • revision a4ff2c53d8  
Tools • Dart 2.18.1 • DevTools 2.15.0  
→ ~ █
```

Announcing



Dart 2.18

ObjC & Swift interop | Platform-specific http | Roadmap

Call directly from Dart to Objective-C and Swift

Support for networking with NSURLSession & Cronet

Important update on null safety

@稀土掘金技术社区

第二章

PART 2

融创数科对 Flutter 的技术积累与使用现状

Sunac Flutter 开发模板

仓库地址: http://gitlab.sunacwy.com.cn/smart-community/sunac-flutter/sunac_flutter_template.git



Crash上报



网络请求



Json&持久化



日志记录



状态管理

GetX

State Manager | Navigation Manager
Dependencies Manager

Fast, Stable, Extra-light and Powerful Flutter Framework

依赖注入

路由导航



常用API

Flutter 适用的异常收集与监控平台

调用栈&行为流

NoSuchMethodError

NoSuchMethodError: The method 'popUntil' was called on null.
Receiver: null
Tried calling: popUntil()

mechanism FlutterError handled true

```
</> pos_visit_questionnaire_page.dart in _PosVisitQuestionnairePageState.build.<fn>.<fn> at line 118 within smt
pos_visit_alert_page.dart in PosVisitAlertPage.build.<fn> at line 83 within smt
Called from: ink_well.dart in _InkResponseState._handleTap within flutter
```

BREADCRUMBS

Filter By

类型	CATEGORY	描述	LEVEL	时间
	navigation	to: / state: didPush	info	08:50:16
	navigation	state: didPush from: /	info	08:50:16
	navigation	to: / state: didPop	info	08:50:17
	navigation	state: didPush from: /	info	08:50:17
	navigation	to: / state: didPop	info	08:50:17
	navigation	to: PageName.home state: didReplace from: /	info	08:50:17
	navigation	state: didPush from: PageName.home	info	08:50:17
	navigation	to: PageName.home	info	08:50:17

快速定位&解决

详细的异常“现场”，还原操作、快速定位并修复问题，即使止损

快速发现&报警

第一时间发现错误或异常，推送邮件或飞书小程序

持续跟踪&评估

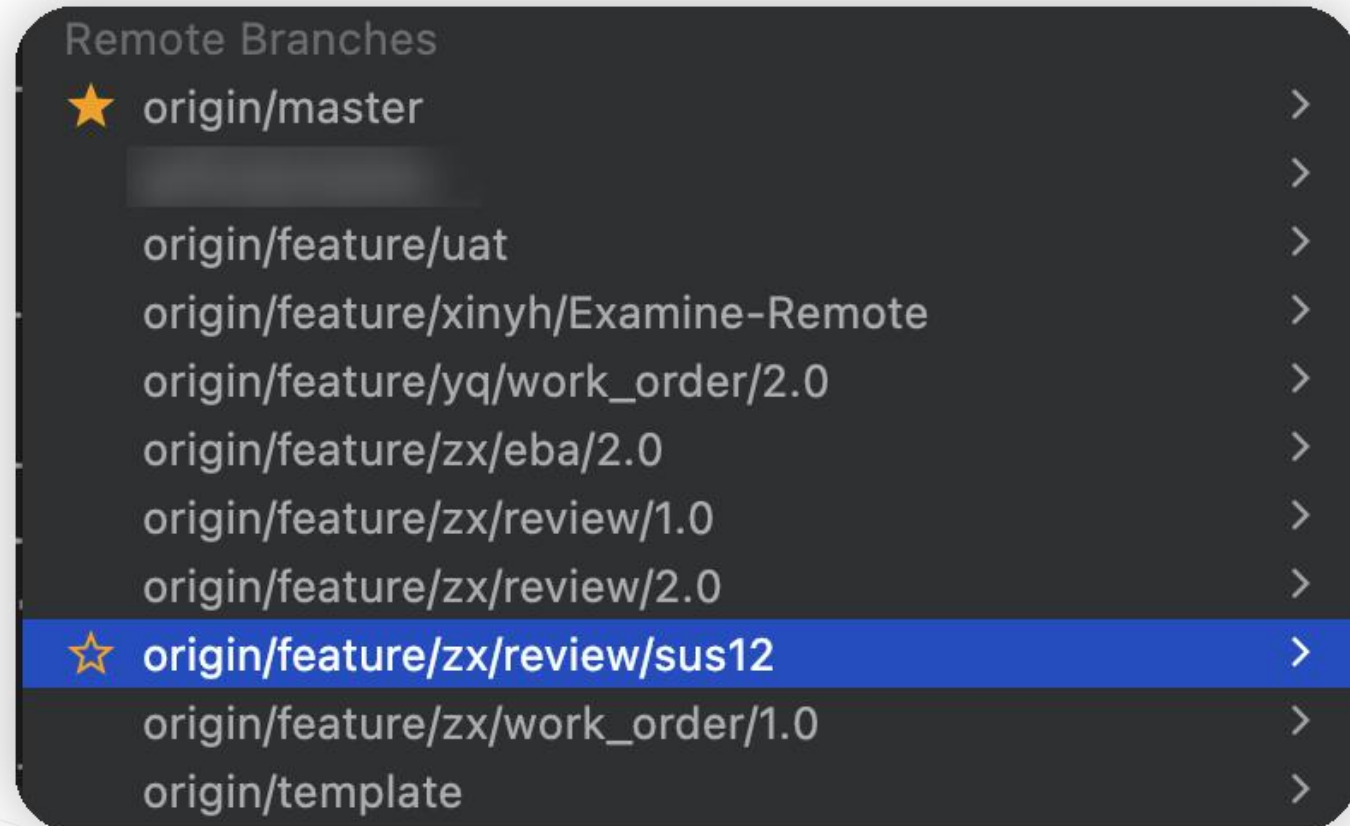
解决问题后，跟踪新版本评估改善

逐步改善

产品的稳定性和用户体验

中国品质服务引领者

Flutter 模板项目与臻心 Flutter 模块项目



自研 Flutter 模板

[sunac_flutter_template](#) 说明文档: [README.md](#)

自研 Flutter 脚手架

[sunac_flutter_cli](#) 说明文档: [README-zh_CN.md](#)

Jenkins Job 参考样例

[Zhenxin-Android](#) (Freestyle)
[Zhenxin-iOS-Flutter-Devops](#) (Pipeline)

学习&研发 参考材料

[Flutter 中文文档](#) [Flutter 官方软件包存储库](#)
[Flutter | 老孟 332个组件大全](#) [GetX架构](#)
[《Flutter实战·第二版》](#)

分享PPT:

- [\(新\) Flutter——为所有屏幕构建精美应用.pdf](#)
- [Flutter——为所有屏幕构建精美应用 纯图版.pdf](#)
- [跨平台开发技术-李斌.pptx](#)

技术文档:

- [臻心Flutter Module 子仓库拉取:依赖拉取:预编译](#)
- [将 Flutter Module 引入现有 归心 Android 应用](#)
- [Android 集成Flutter指南](#)

插件: [Flutter-Toolkit](#) 已使用依赖库: [Flutter](#) 已使用依赖库

项目状态分布

进行中: 1 (25%)



已完成: 3 (75%)

● 已完成
 ● 进行中

《Flutter实战·第二版》

跳转 - 在线文档

使用现状

项目名称	所属条线	使用方式	开发人员	完成进度	项目状态	上线时间	附件及备注
智慧物联1.0 (可视化/轻量化 EBA)	智慧社区	嵌入臻心 APP	@苏晟 @门皓	100%	已完成	2022/06/16	EBA可视化系统PRD文档.docx 智慧物联 (EBA) 系统用户手册 智慧物联EBA (APP) .mp4
智慧物联2.0 (New Feature 能耗)	智慧社区	嵌入臻心 APP	@苏晟 @门皓	100%	已完成	2022/08/08	轻量化EBA系统PRD文档 (RC)
工单投诉	臻心	嵌入臻心 APP	@李斌 @崔少华	100%	已完成	2022/08/30	
融慧1.0 (智慧联动管理平台)	智慧社区	嵌入臻心 APP	@苏晟 @门皓	90%	进行中	2022/09/30	联动管理平台共享资料文件夹 PRD原型
融慧2.0 (现场检查)	智慧社区	嵌入臻心 APP	@辛永辉 @门皓 @苏晟	5%	未开始	2022/11/11	
知识库管理	臻心	嵌入臻心 APP	@李斌 @崔少华	90%	进行中	2022/10/20	

已收藏 最近浏览 全部

UAT大华001
无锡亚美利加项目/1
栋/2单元/3层/401

fake-4891583
无锡亚美利加项目/无
锡亚美利加三期围合...

已收藏 最近浏览 全部

加载失败
刷新重试

UAT大华001
无锡亚美利加项目/1
栋/2单元/3层/401

fake-4891583
无锡亚美利加项目/无
锡亚美利加三期围合...

视频检查

已收藏 最近浏览 全部

全部项目 ▲ 标签 ▼

服务集团 > 晋鲁大区

- 西北业务单元 >
- 太原业务单元 >
- 承德分公司
- 济南业务单元 >
- 烟台分公司

重置 确定

加载失败
刷新重试

UAT大华001
无锡亚美利加项目/1
栋/2单元/3层/401
阿肥发

已收藏

实时 >

抓图报事

远程巡检

全部项目 ▼ 筛选 ▲

计划发起层级

截止日期

日	一	二	三	四	五	六
21	22	23	24	25	26	27
28	29	30	31			

2022年9月

			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

重置 确定

搜索摄像机

搜索摄像机名称

搜索检查任务

任务编号 ▼ 搜索

9:41



现场检查

全部项目 ▾

截止日期 ▲



今天

明天

本周

本月

日	一	二	三	四	五	六
23	23	23	23	23	23	23
23	23	23	23	23	23	23
23	23	23	23			

2021年10月

				23	23	23
23	23	23	23	23	今天	

重置

确定

该检查项已提交过以下报
事，确定提交此新问题吗？

已提交的报事：

JJ类问题：sttsy

工单测试4 2022-09-27 10:38:26

取消

确定提交

保洁 TJ

确认删除吗？

取消

删除

网络错误
请在有网情况下提交

请填写所有必填项

确认已完成任务并提交吗？

提交后任务将为“已完成”状态，不能再增
加或修改“任务详情”、“记录问题”、“报
工单中”的内容

取消

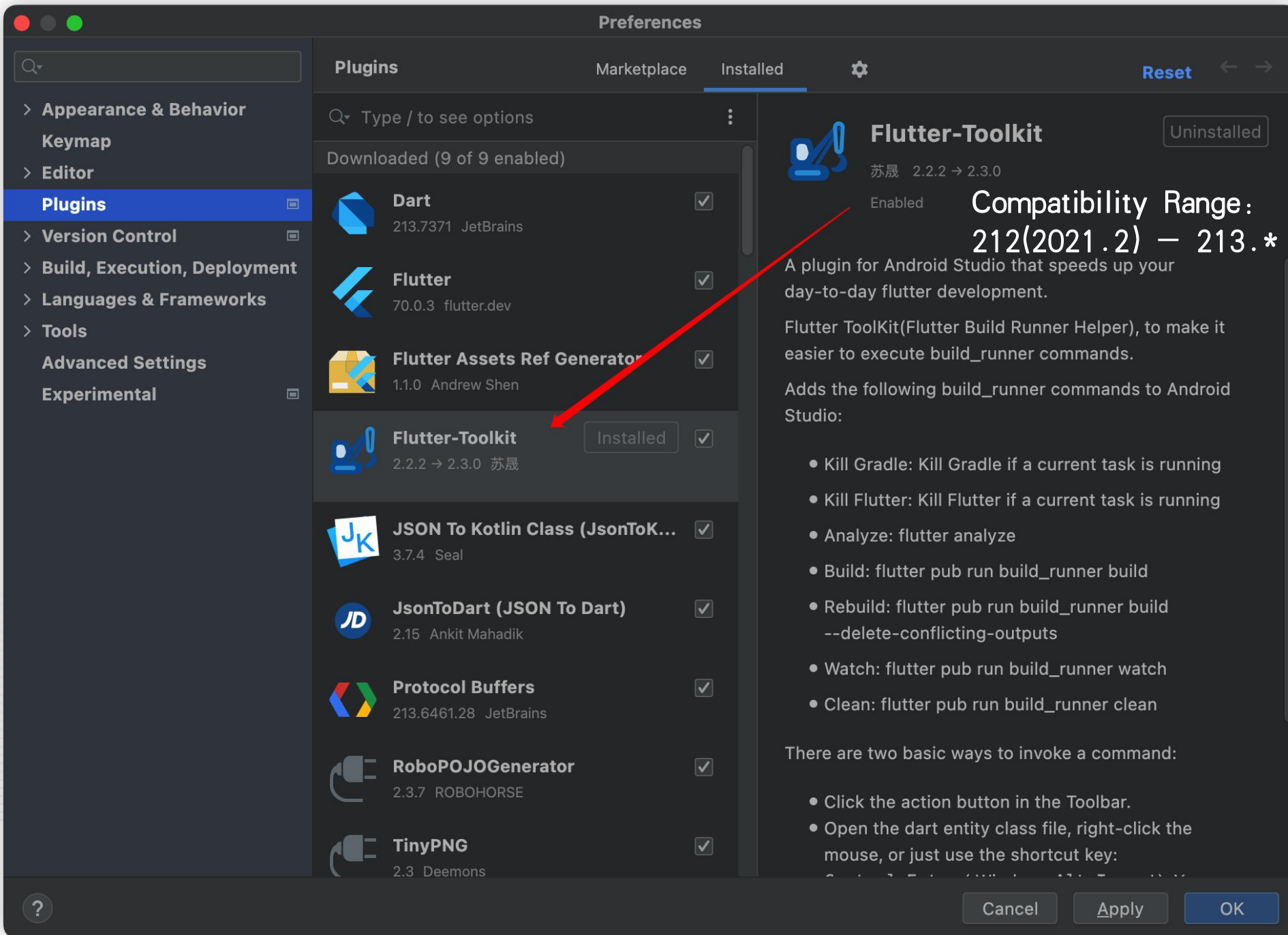
确认提交

✓
已提交

拍照

从相册选择

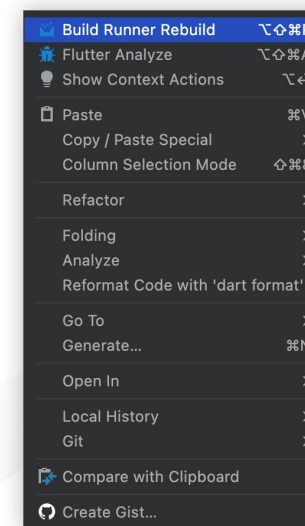
取消



1、无需手动选择 Flutter SDK 路径

2、新增 Flutter Analyze Action

3、支持 *.dart 文件 下右键执行 Action



Flutter 实践路线规划与价值

核心目标：提升人效50%~100%，保障性能和质量，提升提效效果、~~扩大提效范围。~~

人员掌握，业务覆盖
移动开发者逐步掌握Flutter开发
B端新应用/模块/页面评估优先实践
明确持续集成与混合开发方案



核心价值
双端高度一致性，用户体验大幅提升
减少研发/测试人员大量人日
提升开发效率与体验

性能与质量
统一接入 Sunac Flutter 基座/模板
建立异常收集与监控平台，建立性能质量标准
统一状态管理与路由管理方案



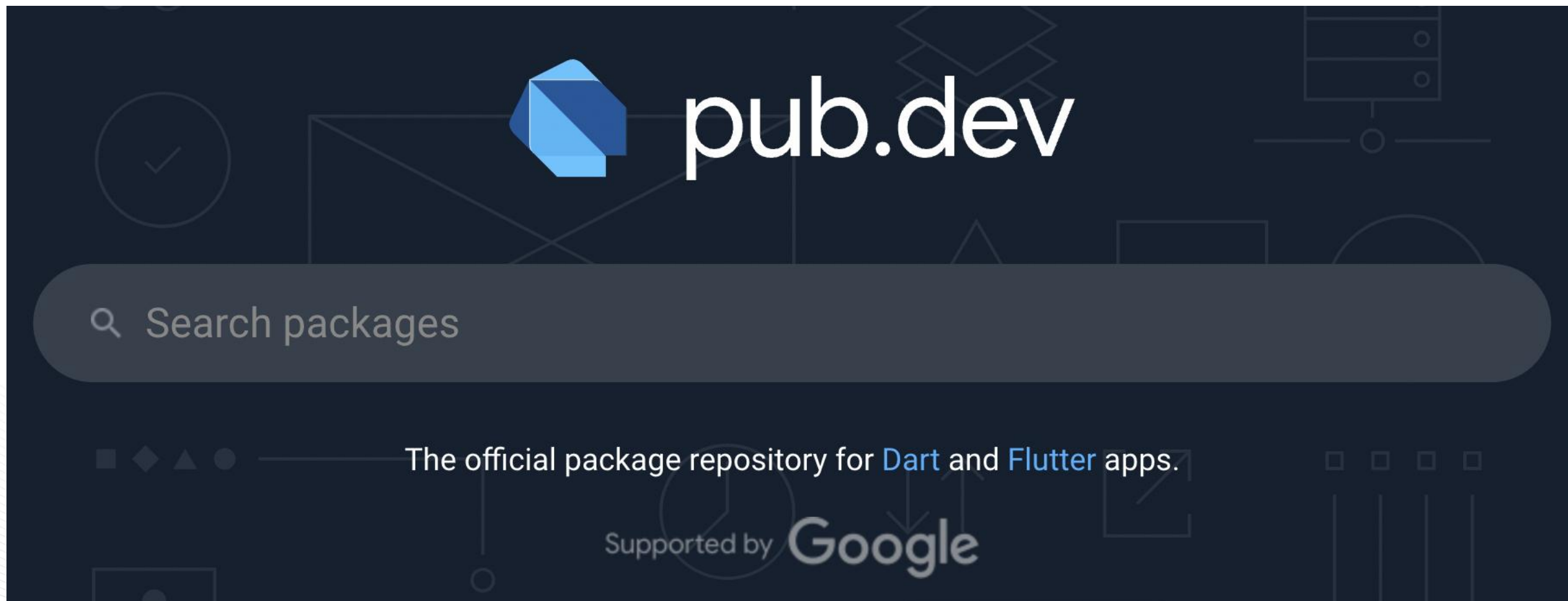
核心价值
减少依赖维护成本，提高风格一致性
增加基建（共享库）价值
提升应用稳定性

第三章

PART 3

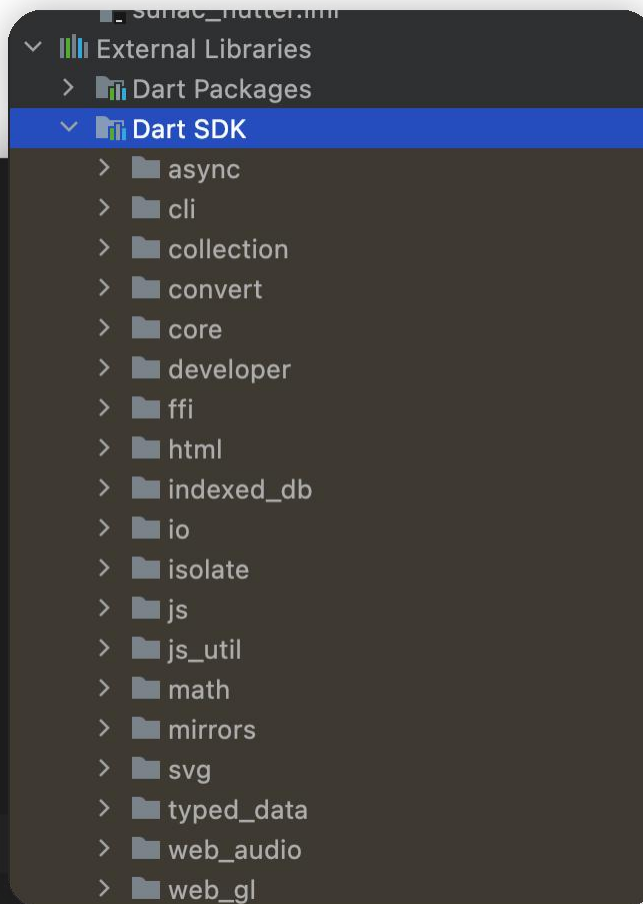
Dart 基本语法与 Flutter 开发实用小技巧

Flutter 的官方软件包存储库 pub.dev



flutter pub deps

```
# Dio network inspector
dio_log : ^2.0.4
# Json generation
json_annotation: ^4.6.0
# Log
logger: ^1.1.0
# Loading/Toast/Notification
bot_toast: ^4.0.3
# Sentry
sentry: ^6.9.1
# Pull to Refresh
pull_to_refresh:
  git:
    url: https://github.com/miquelbeltran/flutter-pulltorefresh
    ref: 6d7fbdd
# Open File(*.xls)
open_file: ^3.2.1
```

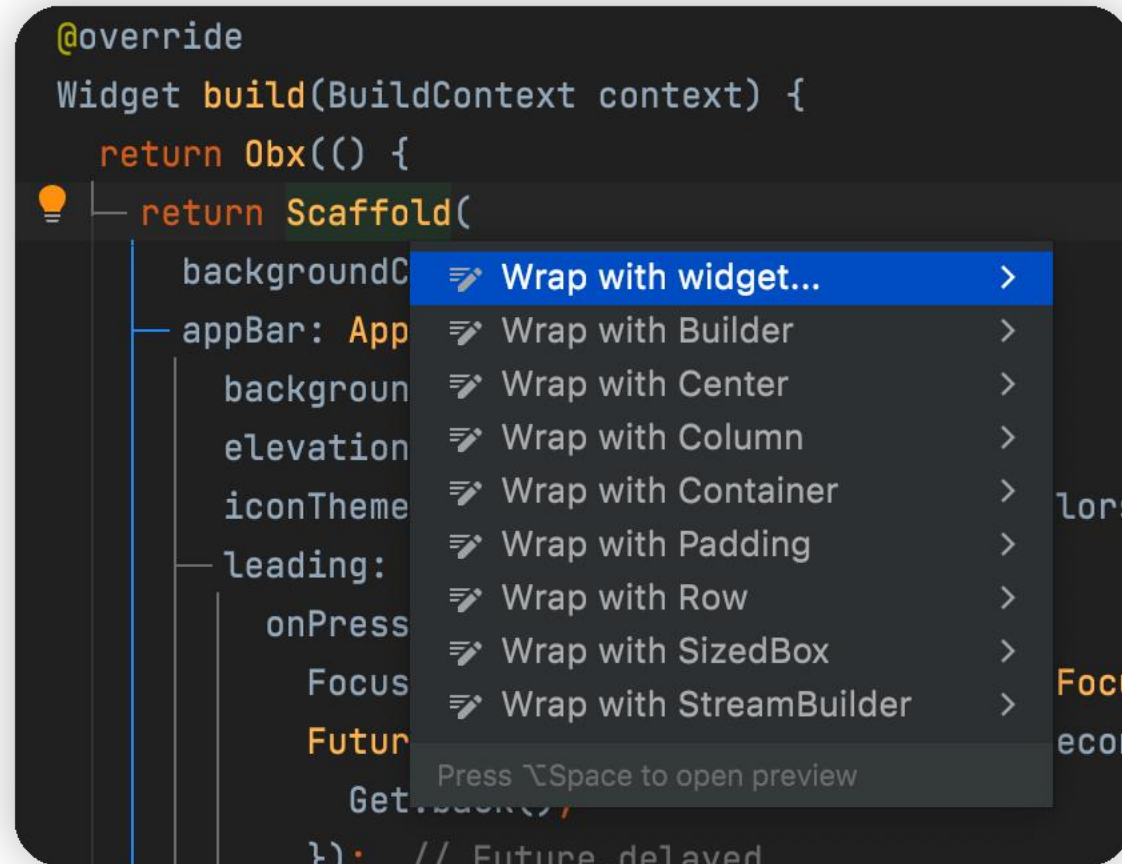


Flutter 已使用依赖库

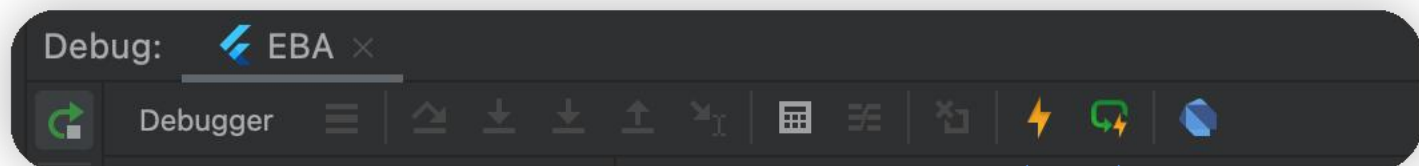
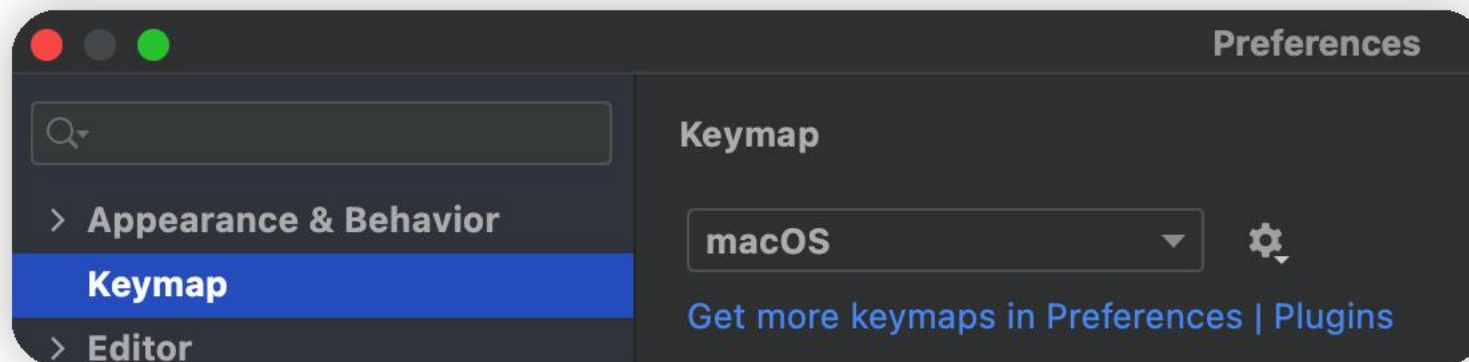
```
sunac_flutter (feature/zx/review/sus12) × flutter pub deps
Dart SDK 2.18.0
Flutter SDK 3.3.0
sunac_flutter 1.0.0+1
|-- bot_toast 4.0.3
|   |-- flutter...
|-- build_runner 2.2.0
|   |-- analyzer 4.7.0
|       |-- _fe_analyzer_shared 47.0.0
|           |-- meta...
|               |-- collection...
|                   |-- convert 3.0.2
|                       |-- typed_data...
|                           |-- crypto...
|                               |-- glob...
|                                   |-- meta...
|                                       |-- package_config...
|                                           |-- path...
|                                               |-- pub_semver...
|                                                   |-- source_span...
|                                                       |-- watcher...
|                                                           |-- yaml...
|-- args 2.3.1
|-- async...
```

Wrap with 小技巧

```
@override
Widget build(BuildContext context) {
  return Obx(() {
    return Scaffold(
      backgroundC
      appBar: App
      backgroun
      elevation
      iconTheme
      leading:
      onPressed
      Focus
      Futur
      Get.back()
    ); // Future delayed
  });
}
```



Android Studio Flutter常用快捷键



Flutter Hot Reload
⌘+\

Flutter Hot Restart
⇧+⌘+\

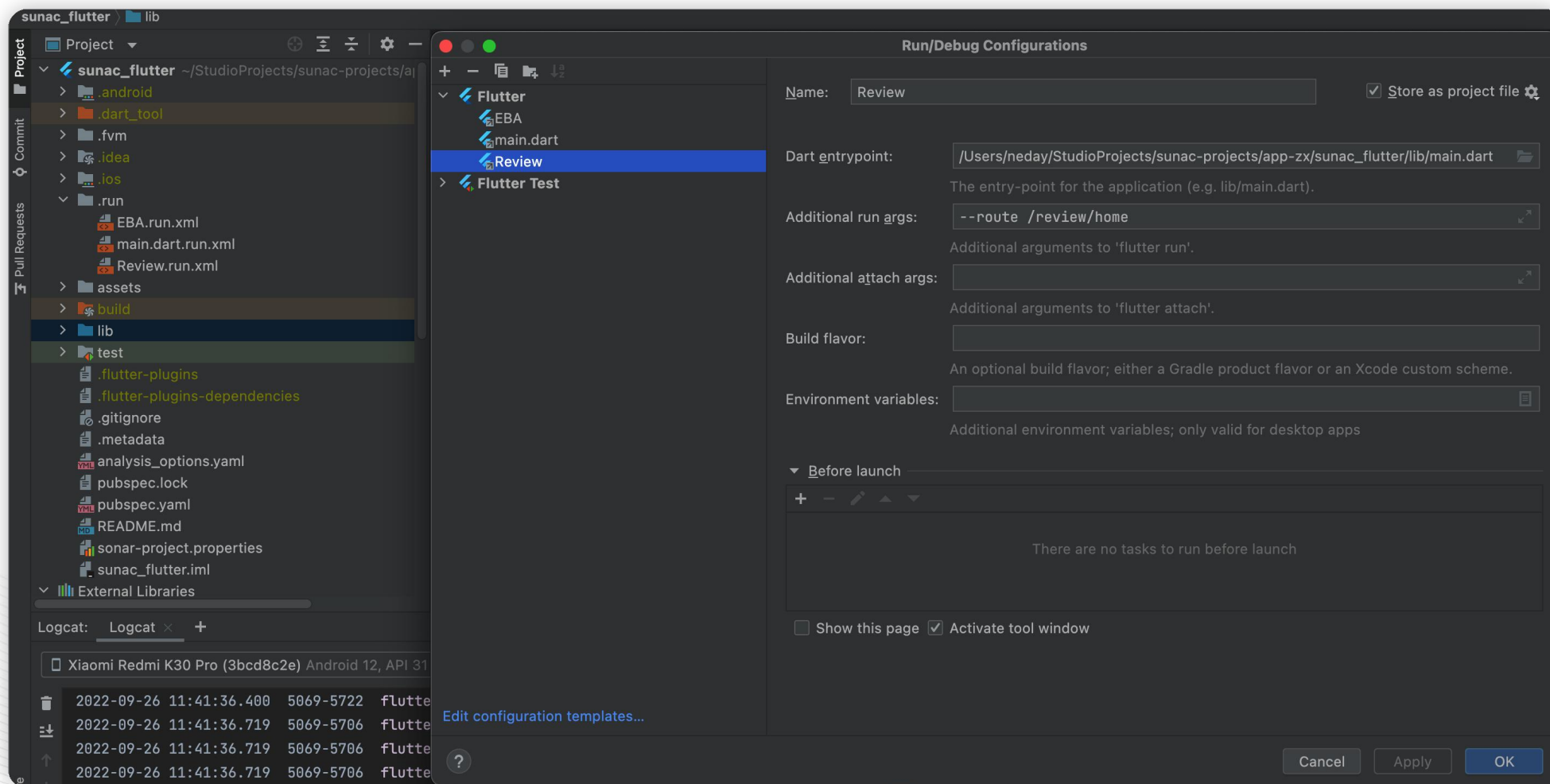
Restart Dart Analysis Server

智能提醒: ⌘+⌘

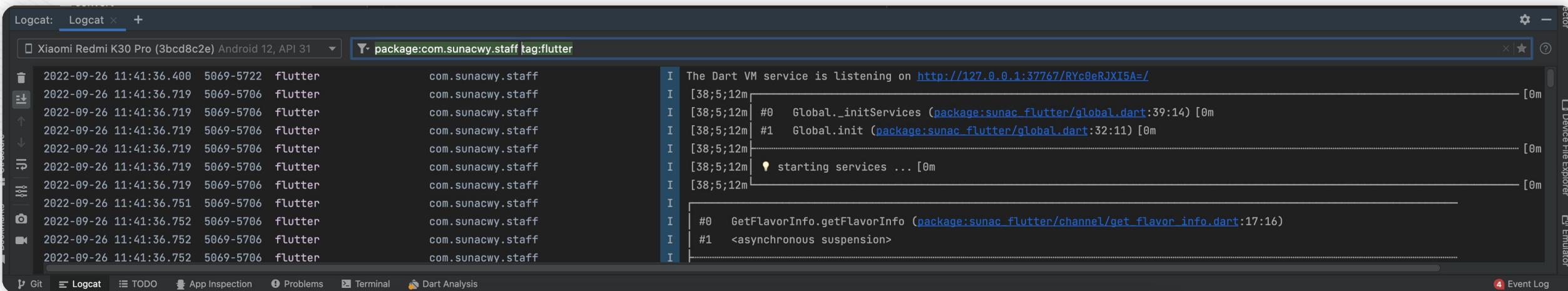
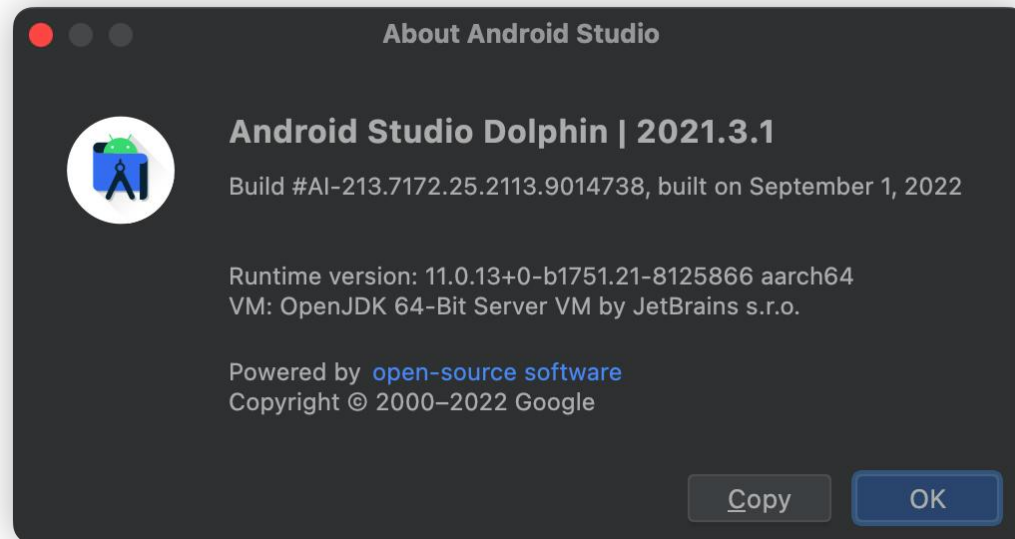
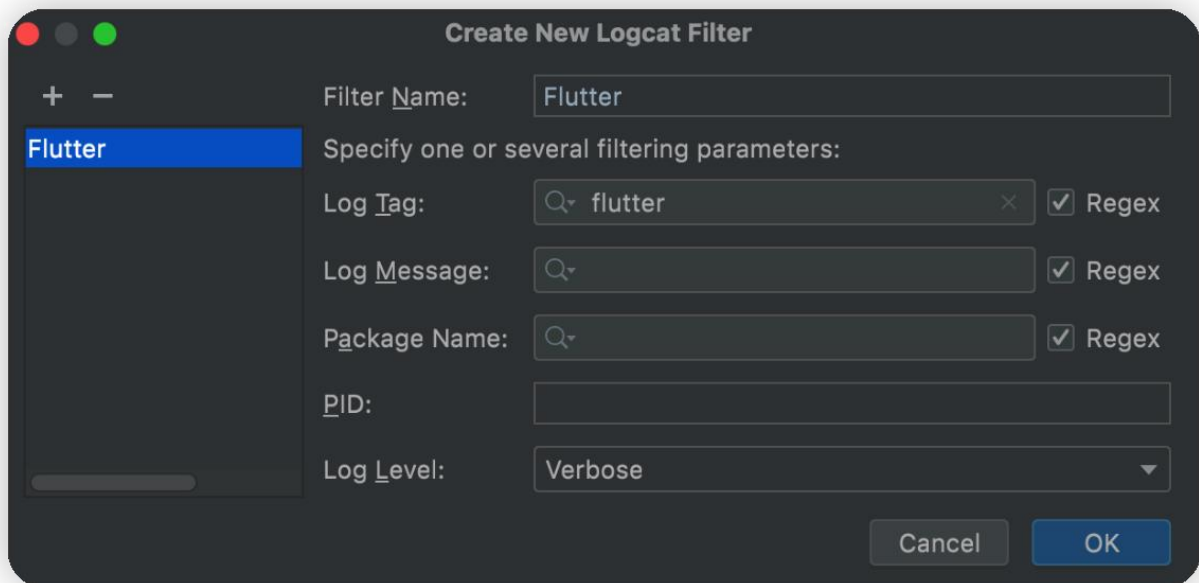
批量重命名: ⇧+F6

全文格式化: ⌘+⇧+L

Flutter 路由入口 小技巧



Flutter Logcat Filter



Flutter 常用 Widget

在 Flutter 中, 万物皆 Widget

MaterialApp

Scaffold

AppBar

BottomNavigation
Bar

Container

Padding

SizedBox

Flex
Column/Row

Expanded

Wrap

Align/Center

Stack
IndexedStack

Text

SingleChildScroll
View

ListView
GridView

WillPopScope

Dart 语言简介 —— 变量声明

var
dynamic&Object
final&const

```
1 var t = "hi world";
2 // 下面代码在dart中会报错，因为变量t的类型已经确定为String，
3 // 类型一旦确定后则不能再更改其类型。
4 t = 1000;
```

```
1 dynamic t;
2 Object x;
3 t = "hi world";
4 x = 'Hello Object';
5 //下面代码没有问题
6 t = 1000;
7 x = 1000;
```

```
1 dynamic a;
2 Object b = "";
3 main() {
4     a = "";
5     printLengths();
6 }
7
8 printLengths() {
9     // 正常
10    print(a.length);
11    // 报错 The getter 'length' is not defined for the class 'Object'
12    print(b.length);
13 }
```

```
1 //可以省略String这个类型声明
2 final str = "hi world";
3 //final String str = "hi world";
4 const str1 = "hi world";
5 //const String str1 = "hi world";
```

Dart 语言简介 —— 空安全 (null-safety)

```

1  test() {
2      int i;
3      print(i*8);
4  }
```

```

1  int i = 8; //默认为不可空, 必须在定义时初始化。
2  int? j; // 定义为可空类型, 对于可空变量, 我们在使用前必须判空。
3
4  // 如果我们预期变量不能为空, 但在定义时不能确定其初始值, 则可以加上late关键字,
5  // 表示会稍后初始化, 但是在正式使用它之前必须得保证初始化过了, 否则会报错
6  late int k;
7  k=9;
```

```

1  class Test{
2      int? i;
3      Function? fun;
4      say(){
5          if(i!=null) {
6              print(i! * 8);
7          }
8          if(fun!=null){
9              fun!(); // 同上
10         }
11     }
12 }
```

```
1  fun?.call()
```


Dart 语言简介 —— 函数

```
1 bool isNoble(int atomicNumber) {
2     return _nobleGases[atomicNumber] != null;
3 }
```

```
1 var say = (str){
2     print(str);
3 };
4 say("hi world");
```

```
1 typedef bool CALLBACK();
2
3 //不指定返回类型, 此时默认为dynamic, 不是bool
4 isNoble(int atomicNumber) {
5     return _nobleGases[atomicNumber] != null;
6 }
7
8 void test(CALLBACK cb){
9     print(cb());
10 }
11 //报错, isNoble不是bool类型
12 test(isNoble);
```

```
1 bool isNoble (int atomicNumber)=> true ;
```

```
1 void execute(var callback) {
2     callback();
3 }
4 execute(() => print("xxx"))
```

Dart 语言简介 —— 参数

```

1 String say(String from, String msg, [String? device]) {
2     var result = '$from says $msg';
3     if (device != null) {
4         result = '$result with a $device';
5     }
6     return result;
7 }

```

```

1 //设置[bold]和[hidden]标志
2 void enableFlags({bool bold, bool hidden}) {
3     // ...
4 }

```

```
1 say('Bob', 'Howdy'); //结果是: Bob says Howdy
```

```
1 enableFlags(bold: true, hidden: false);
```

```
1 say('Bob', 'Howdy', 'smoke signal'); //结果是: Bob says Howdy with a smoke signal
```

Dart 语言简介 —— mixin

```

1  class Person {
2      say() {
3          print('say');
4      }
5  }

```

```

25  class Dog with Eat, Walk{}
26  class Man extends Person with Eat, Walk, Code{}

```

```

7  mixin Eat {
8      eat() {
9          print('eat');
10     }
11 }
12
13 mixin Walk {
14     walk() {
15         print('walk');
16     }
17 }
18
19 mixin Code {
20     code() {
21         print('key');
22     }
23 }

```

Dart 语言简介 —— mixin

```

1  class Person {
2      say() {
3          print('say');
4      }
5  }

```

```

25  class Dog with Eat, Walk{}
26  class Man extends Person with Eat, Walk, Code{}

```

```

7  mixin Eat {
8      eat() {
9          print('eat');
10     }
11 }

12

13 mixin Walk {
14     walk() {
15         print('walk');
16     }
17 }

18

19 mixin Code {
20     code() {
21         print('key');
22     }
23 }

```

Dart 语言简介 —— 异步支持

```
1 Future.delayed(Duration(seconds: 2), () {  
2     //return "hi world!";  
3     throw AssertionError("Error");  
4 }).then((data) {  
5     //执行成功会走到这里  
6     print(data);  
7 }).catchError((e) {  
8     //执行失败会走到这里  
9     print(e);  
10 }).whenComplete(() {  
11     //无论成功或失败都会走到这里  
12 });
```

```
1 login("alice","*****").then((id){  
2     return getUserInfo(id);  
3 }).then((userInfo){  
4     return saveUserInfo(userInfo);  
5 }).then((e){  
6     //执行接下来的操作  
7 }).catchError((e){  
8     //错误处理  
9     print(e);  
10 });
```

```
1 task() async {  
2     try{  
3         String id = await login("alice","*****");  
4         String userInfo = await getUserInfo(id);  
5         await saveUserInfo(userInfo);  
6         //执行接下来的操作  
7     } catch(e){  
8         //错误处理  
9         print(e);  
10    }  
11 }
```

Dart 常见 关键词

if else
for/for in
forEach

while
do while

break
continue

switch/case
default

assert

abstract

implements

extend & super

dynamic

import
show & hide

as, is, is!

static
factory

async & await

export

library & part

mixin & with &
on



Let's Flutter!

SUNAC 融创服务
至 善 · 致 美

感谢聆听

SUNAC SERVICES

中国品质服务引领者