

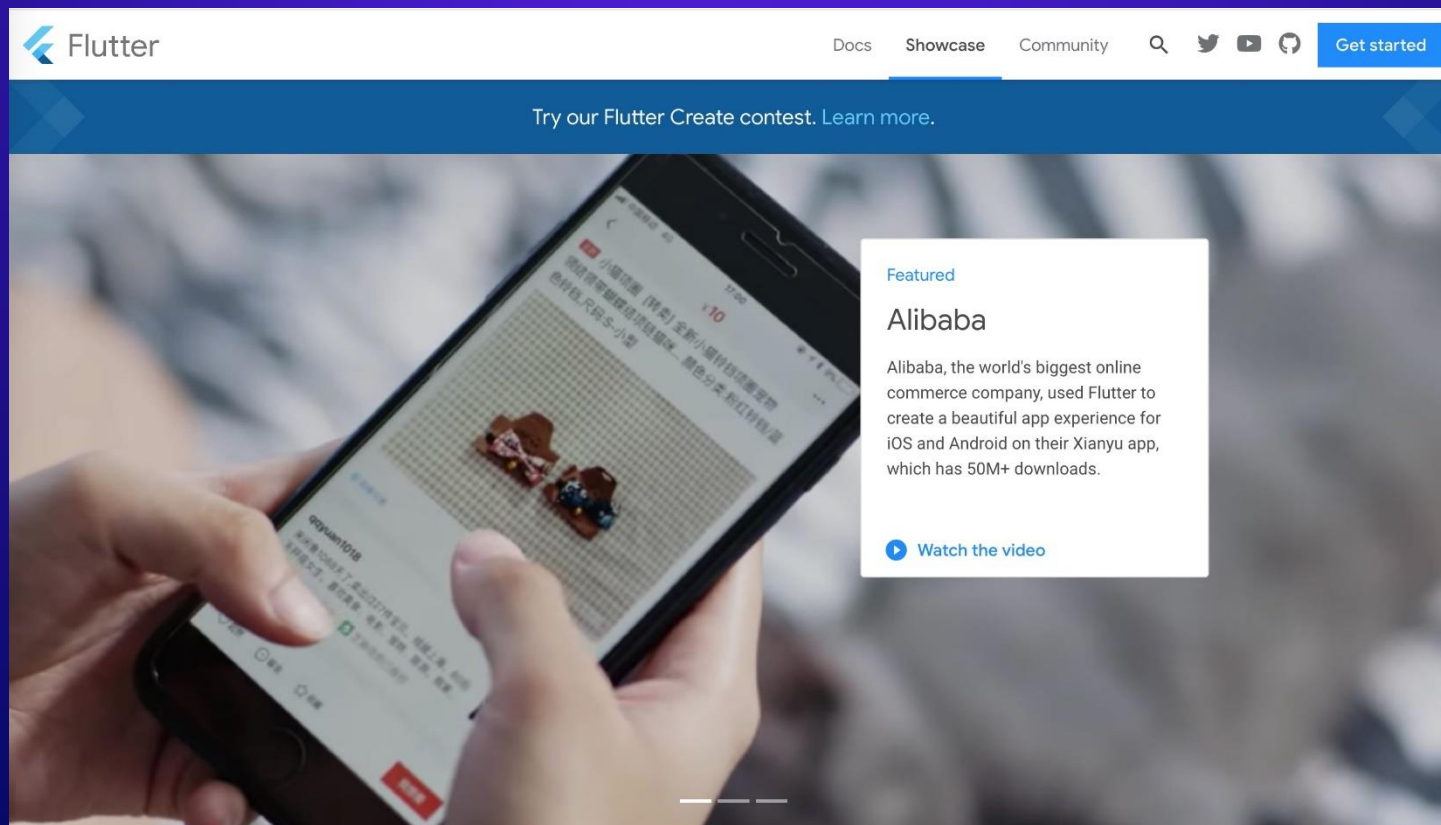


APKBUS

—
Flutter-闲鱼的探索与收获

熊华丽

</> 闲鱼的实践



前期技术认证

技术体系融合

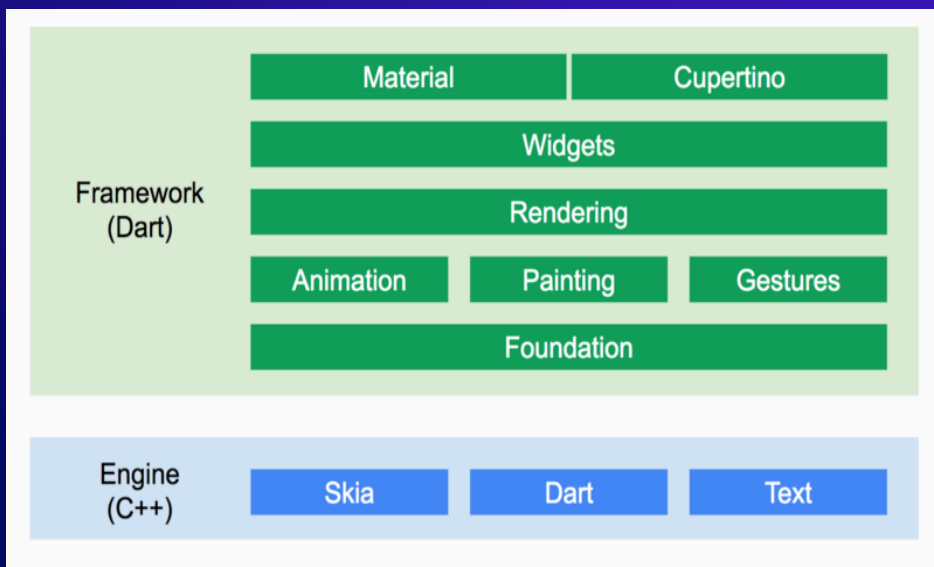
基础设施建设

</> 我需要flutter吗？

Flutter能带来什么？

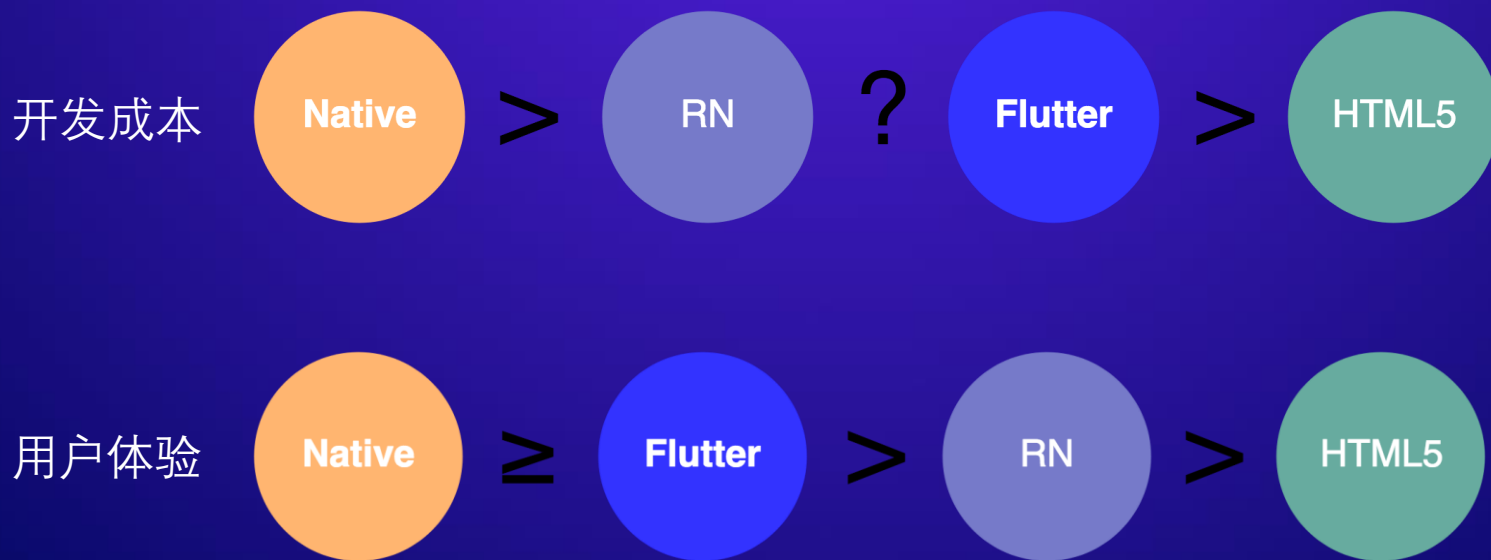


</> Flutter的理论优势



仅仅借用平台原生的图形渲染能力，使用定制的运行引擎和UI Widget，实现更彻底的跨平台，并能在技术上做深度优化，获得更好的开发效率和运行性能。

</> 闲鱼的实践结论



Flutter适合有跨栈需求，并且需要良好用户体验的业务。

</> 闲鱼的选择



兼容使用多套技术栈，
在具体业务场景选择最合适的技术栈，
充分利用各个技术栈的优势，
各取所长，相互补充。

</> 选择Flutter



</> 技术融合中的问题

1.



2.

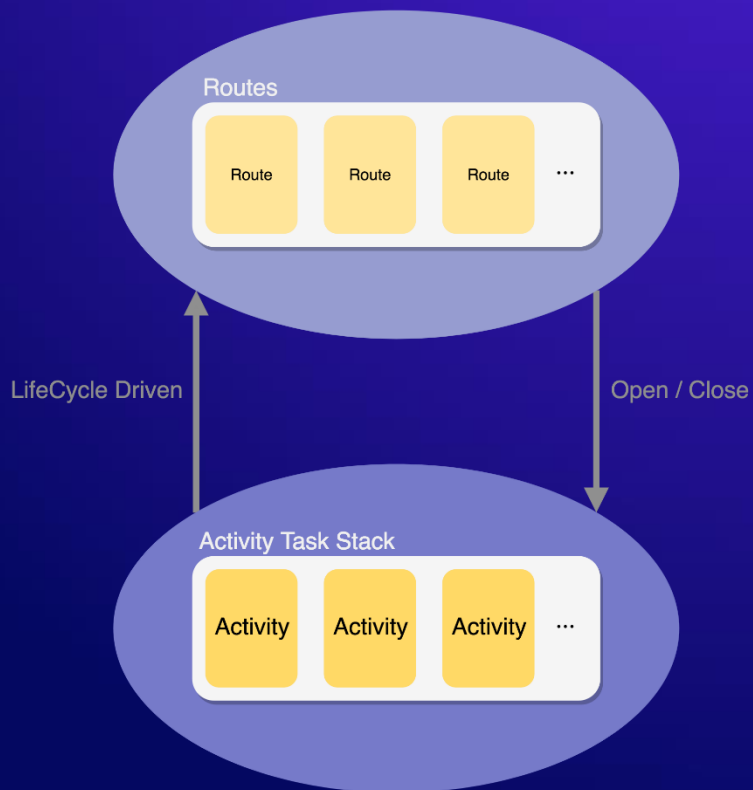


</> 混合栈



- 将Flutter页面无差别的接入到native页面路由系统之中。
- 补充Flutter的嵌入开发能力。
- 封装复杂页面栈的调度逻辑并进行优化。

</> Flutter-Boost



使用原生页面作为容器，并和Flutter页面建立一一映射关系，基于平台原生页面生命周期去驱动Flutter页面生命周期。

</> Flutter-Boost 实现难点



</> Code Show

```
class MyApp extends StatefulWidget {  
  @override  
  _MyAppState createState() => _MyAppState();  
}  
  
class _MyAppState extends State<MyApp> {  
  @override  
  void initState() {  
    super.initState();  
  
    ///页面路由注册  
    FlutterBoost.singleton.registerPageBuilders({  
      'first': (pageName, params, _) => FirstRouteWidget(),  
      'second': (pageName, params, _) => SecondRouteWidget(),  
    });  
  
    ///栈顶页询问加载  
    FlutterBoost.handleOnStartPage();  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Flutter Demo',  
      builder: FlutterBoost.init(),  
      home: Container();  
    );  
  }  
}
```

关键

</> 业务同学的苦恼

组件边界模糊，使用全局作用域

代码千人千面，项目交接困难

以前的工作不能沉淀复用



</> 需要一个业务编程框架



建立规范，达成共识，提供基础能力。

</> 社区的方案

Scoped_Model

- 简单，上手容易
- 基础的监听者模型

BLoC

- 响应式流式编程
- 清晰的数据流

Flutter-Redux

- 强大的状态管理能力
- 逻辑界面彻底解耦

应对复杂场景能力 / 使用成本

低

高

</> 闲鱼的需求

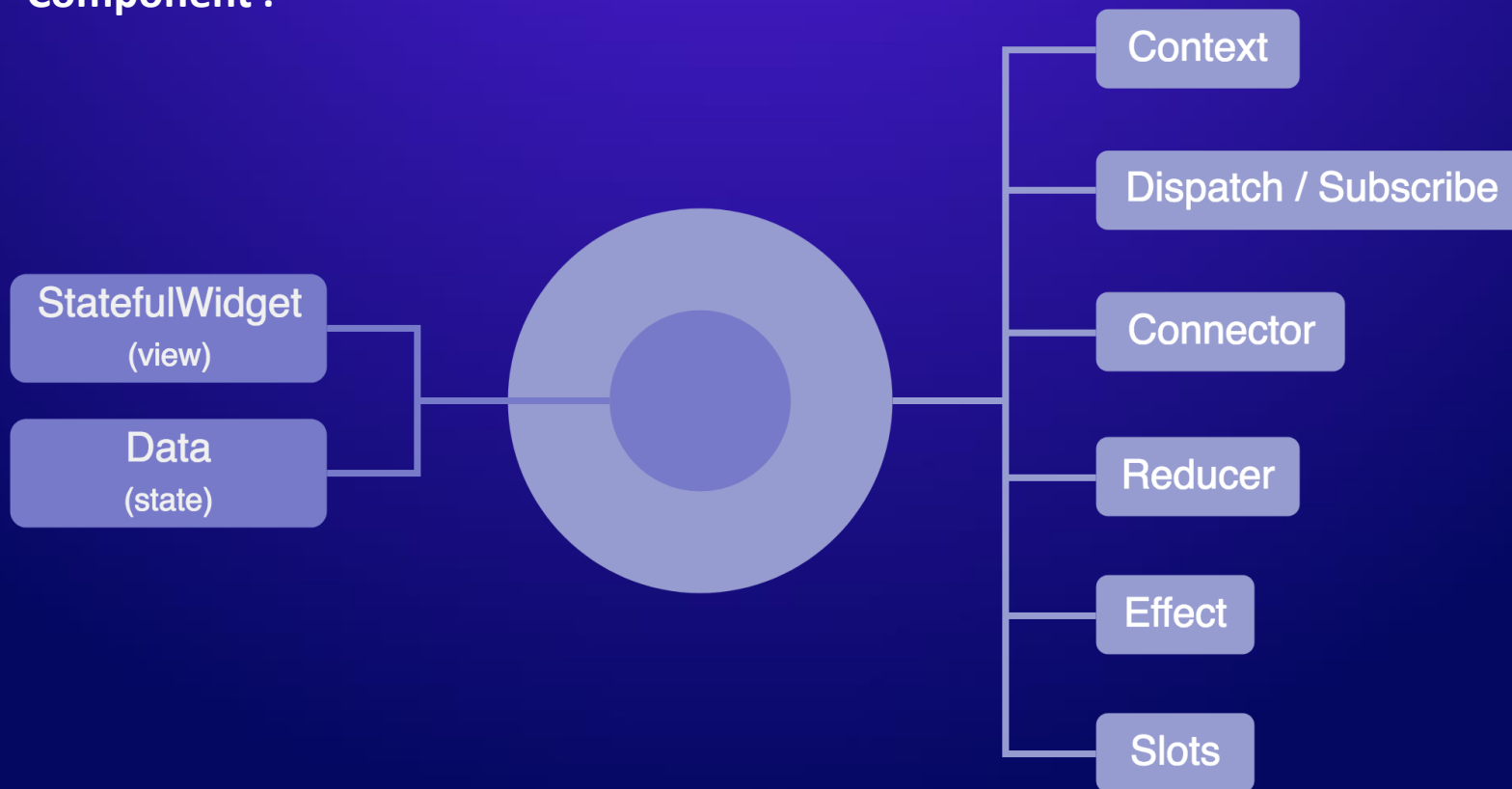


`</>` fish-redux



</> 最小的业务单元

Component :



</> 应用举例和效果分析



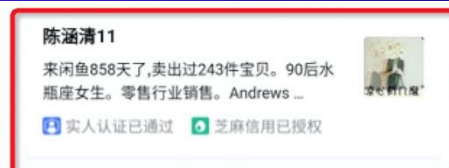
视频 →



卖家信息 →



详情描述 →



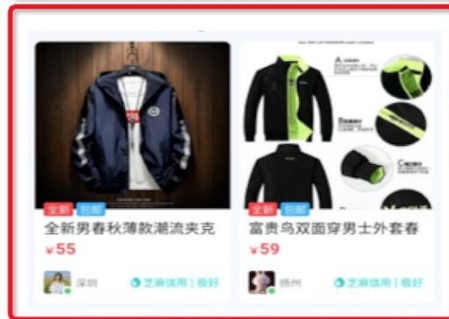
← 闲鱼号



← 留言



← 鱼塘



← 推荐

</> Code Show

```
▼ detail_container
  action.dart
  effect.dart
  page.dart
  reducer.dart
  state.dart
  view.dart

10 class DetailContainer extends Page<DetailContainerState, Object> {
11   DetailContainer()
12     : super(
13       initState: initState,
14       view: buildMainView,
15       effect: DetailContainerEffectBuilder.build(),
16       reducer: asReducer(DetailContainerReducerBuilder.buildMap()),
17       middlewares: <Middleware<DetailContainerState>>[
18         performanceMiddleware<DetailContainerState>(
```

```
▼ app_bar_component
  action.dart
  component.dart
  effect.dart
  reducer.dart
  state.dart
  view.dart

12 class AppBarComponent extends Component<AppBarState> {
13   AppBarComponent()
14     : super(
15       view: AppBarView().asView(),
16       higherEffect: higherEffect(() => AppBarEffectPart()),
17       reducer: asReducer(AppBarReducerBuilder.buildMap()),
18     );
19 }
```

</> 上线前的忐忑

提测前：



测试后：



上线后：



</> 建立开发数据闭环



数据是信仰，No Data No BB！

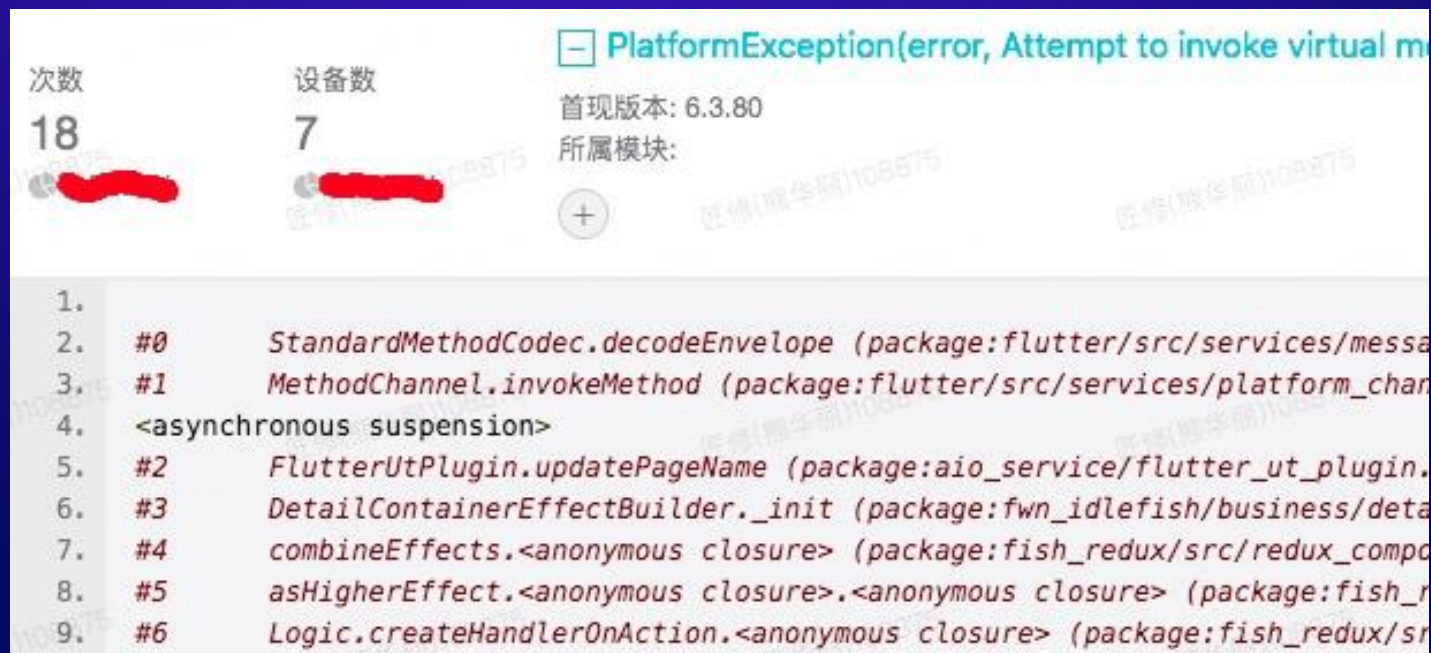
—— 鲁迅

</> 稳定性监控

数据大盘：



异常详情：



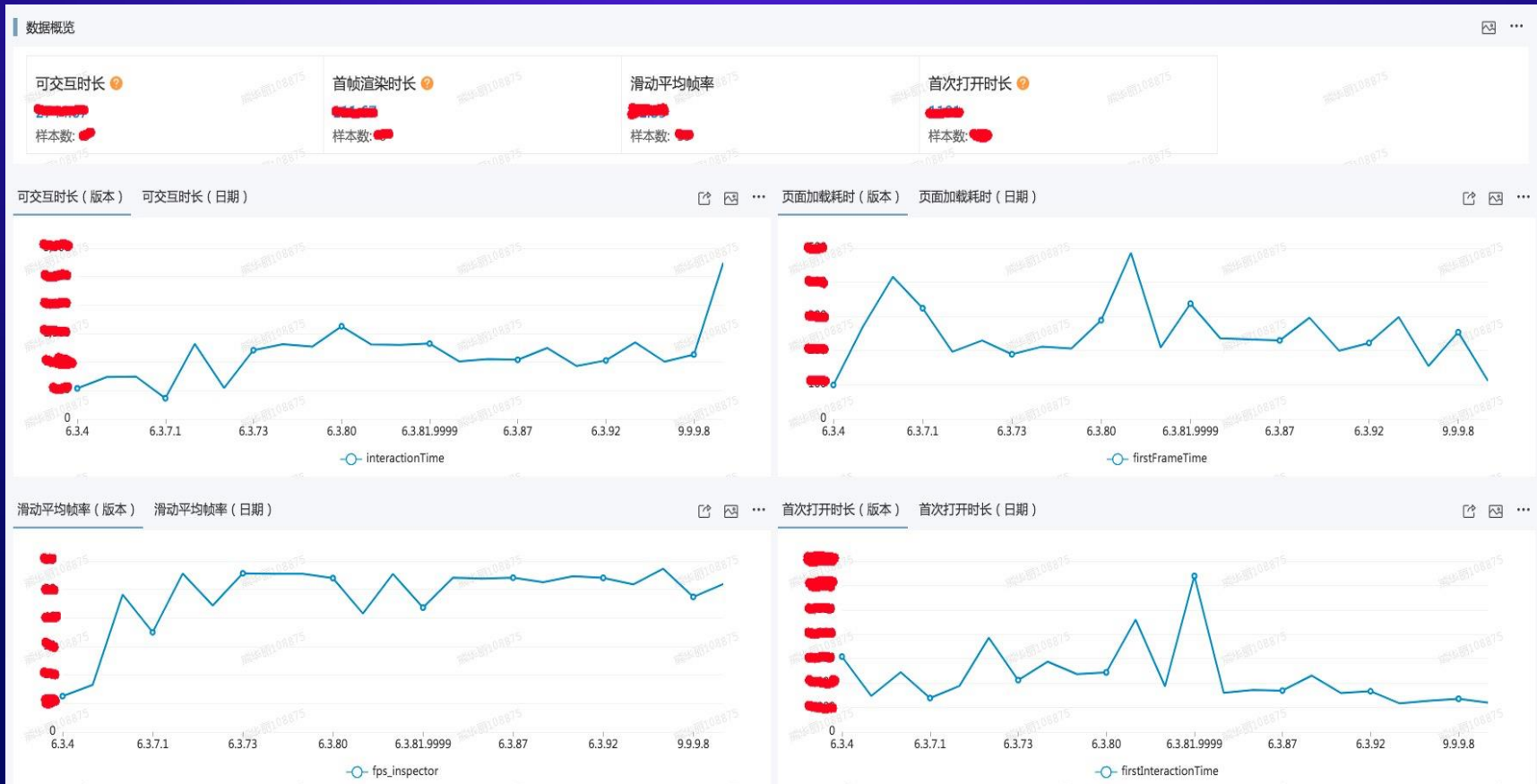
</> Code Show

```
FlutterError.onError = (FlutterErrorDetails details,  
    {bool forceReport: false}) {  
    FlutterError.dumpErrorToConsole(details, forceReport: forceReport);  
    ///add report code here...  
};
```

```
void addIsolateExceptionListener() {  
    Isolate.current.addErrorListener(RawReceivePort((dynamic pair) async {  
        ///report exception such as:  
        // await singleton.reportError(  
        //     (pair as List<String>).first, (pair as List<String>).last, "ERROR");  
    }).sendPort);  
}
```

```
void main() async {  
    runZoned<Future<Null>>(() async {  
        ///some code here  
    }, onError: (dynamic error, dynamic stackTrace) async {  
        ///report error  
    });  
}
```


</> 性能监控



</> Code Show

```
class FXWidgetsFlutterBinding extends WidgetsFlutterBinding {  
  
  @override  
  ImageCache createImageCache() {  
    ///可以自定义图片缓存  
    return super.createImageCache();  
  }  
  
  @override  
  void handleBeginFrame(Duration rawTimeStamp) {  
    ///帧绘制开始  
    super.handleBeginFrame(rawTimeStamp);  
  }  
  
  @override  
  void handleDrawFrame() {  
    ///帧绘制结束  
    super.handleDrawFrame();  
  }  
  
  @override  
  void dispatchEvent(PointerEvent event, HitTestResult result) {  
    ///事件处理  
    super.dispatchEvent(event, result);  
  }  
}
```

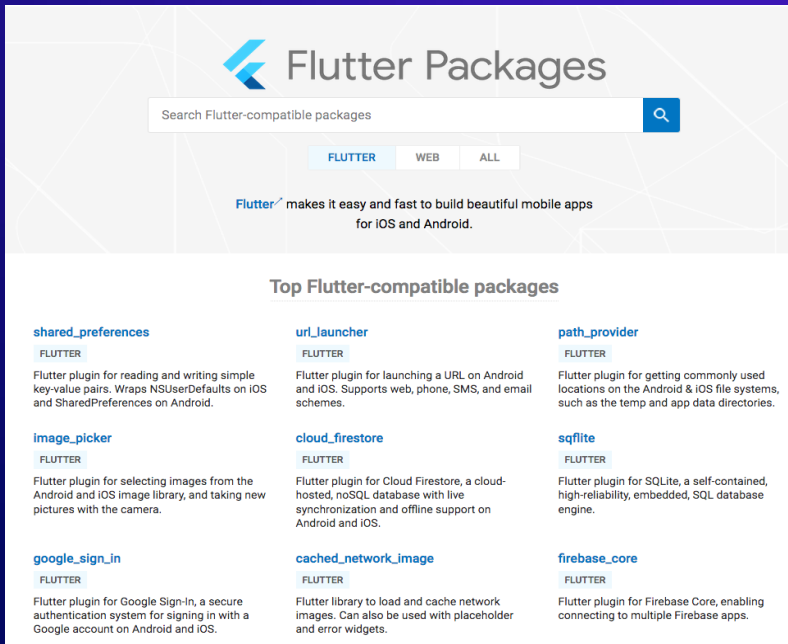
</> Flutter好搞吗？

成本是不是太高？，踩到坑怎么办？



</> pub 和 github

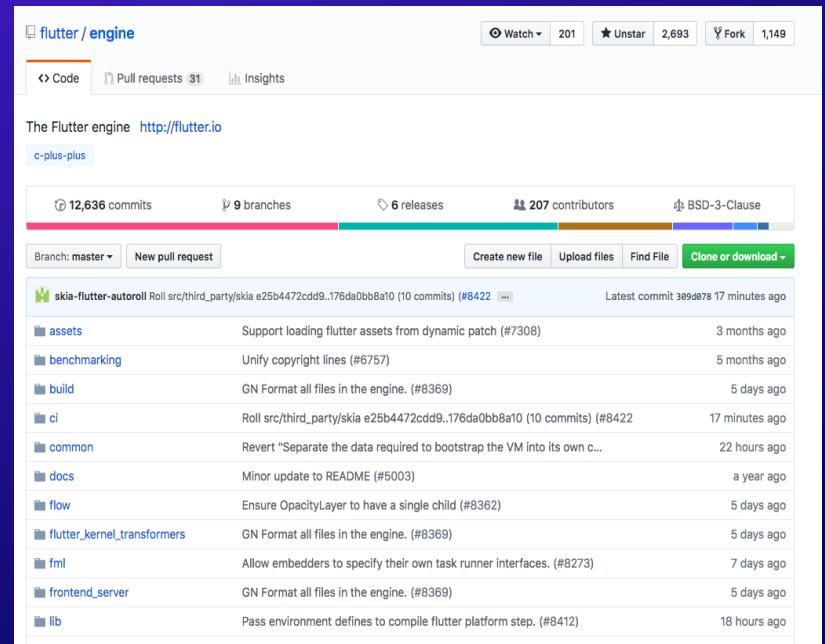
<https://pub.dartlang.org/flutter>



The screenshot shows the Flutter Packages website. At the top, there's a search bar labeled "Search Flutter-compatible packages" and a "Flutter Packages" header. Below the search bar, there are tabs for "FLUTTER", "WEB", and "ALL". A tagline states: "Flutter makes it easy and fast to build beautiful mobile apps for iOS and Android." The main section is titled "Top Flutter-compatible packages" and lists several packages in a grid:

- shared_preferences** (FLUTTER): Flutter plugin for reading and writing simple key-value pairs. Wraps NSUserDefaults on iOS and SharedPreferences on Android.
- url_launcher** (FLUTTER): Flutter plugin for launching a URL on Android and iOS. Supports web, phone, SMS, and email schemes.
- path_provider** (FLUTTER): Flutter plugin for getting commonly used locations on the Android & iOS file systems, such as the temp and app data directories.
- image_picker** (FLUTTER): Flutter plugin for selecting images from the Android and iOS image library, and taking new pictures with the camera.
- cloud_firestore** (FLUTTER): Flutter plugin for Cloud Firestore, a cloud-hosted, noSQL database with live synchronization and offline support on Android and iOS.
- sqflite** (FLUTTER): Flutter plugin for SQLite, a self-contained, high-reliability, embedded, SQL database engine.
- google_sign_in** (FLUTTER): Flutter plugin for Google Sign-In, a secure authentication system for signing in with a Google account on Android and iOS.
- cached_network_image** (FLUTTER): Flutter library to load and cache network images. Can also be used with placeholder and error widgets.
- firebase_core** (FLUTTER): Flutter plugin for Firebase Core, enabling connecting to multiple Firebase apps.

<https://github.com/flutter/engine>



The screenshot shows the GitHub repository for the Flutter engine. The repository name is "flutter / engine". It has 201 watchers, 2,693 stars, and 1,149 forks. The repository is under the "Code" tab, showing 12,636 commits, 9 branches, 6 releases, 207 contributors, and a BSD-3-Clause license. The latest commit is by "skia-flutter-autoroll" 17 minutes ago. Below the repository information, there is a table of files and their commit history:

File	Commit Message	Time Ago
assets	Support loading flutter assets from dynamic patch (#7308)	3 months ago
benchmarking	Unify copyright lines (#6757)	5 months ago
build	GN Format all files in the engine. (#8369)	5 days ago
ci	Roll src/third_party/skia e25b4472cdd9..176da0bb8a10 (10 commits) (#8422)	17 minutes ago
common	Revert "Separate the data required to bootstrap the VM into its own c..."	22 hours ago
docs	Minor update to README (#5003)	a year ago
flow	Ensure OpacityLayer to have a single child (#8362)	5 days ago
flutter_kernel_transformers	GN Format all files in the engine. (#8369)	5 days ago
fml	Allow embedders to specify their own task runner interfaces. (#8273)	7 days ago
frontend_server	GN Format all files in the engine. (#8369)	5 days ago
lib	Pass environment defines to compile flutter platform step. (#8412)	18 hours ago

</> 闲鱼的开源

fish_serializable

JSON序列化和反序列化

flutter_boost

混合栈

Fish_Redux

高效的业务
开发框架

More

On the way...

annotation_route

注解处理

</> 关注我们



THANKS

</> 欢迎关注安卓巴士公众号



www.apkbus.com