

Flutter 异常捕获方式

- 同步异常捕获

同步捕获：使用 `try catch`

```
try {  
    throw FlutterError('同步异常');  
} catch (e) {  
    print(e);  
}
```

- 异步异常捕获

方法 1:

使用 `Flutter` 提供的 `catchError` 捕获

```
Future.delayed(Duration(seconds: 1)).then((value) {  
    throw FlutterError('异步异常');  
}).catchError((e) {  
    print(e);  
});
```

方法 2:

将异步转为同步捕获

```
try {
```

```

        await Future.delayed(Duration(seconds: 1), () {
            throw FlutterError('异步');
        });
    } catch (e) {
        print(e);
    }
}

```

• 全局捕获

方法 1:

在入口函数 `main` 里全局捕获

```

void main() {
    HiDefend.run(MyApp());
}

import 'dart:async';

import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:hi_base/log_util.dart';

class HiDefend {
    /// 全局异常捕获
    static run(Widget app) {
        /// 捕获flutter框架异常
        FlutterError.onError = (FlutterErrorDetails details) {
            if (kReleaseMode) {

```

```

        // 线上环境，走上报逻辑
        Zone.current.handleUncaughtError(details.exception,
details.stack!);
    } else {
        // 开发环境，走console抛出
        FlutterError.dumpErrorToConsole(details);
    }
};

/// 捕获程序异常
runZonedGuarded(() {
    runApp(app);
}, (e, s) => _reportError(e, s));
}

/// 通过接口上报异常
static _reportError(Object e, StackTrace s) {
    LogUtil.log('error', e);
    LogUtil.log('StackTrace', s);
}
}

```

方法2:

使用 `bugly` 的 `flutter` 插件捕获

```

dependencies:
  flutter_bugly: ^0.3.3

```

