

一、简介

二、使用

1、使用

2、使用BuildContext

3、将SnackBar的Widget拆分出来

4、截图

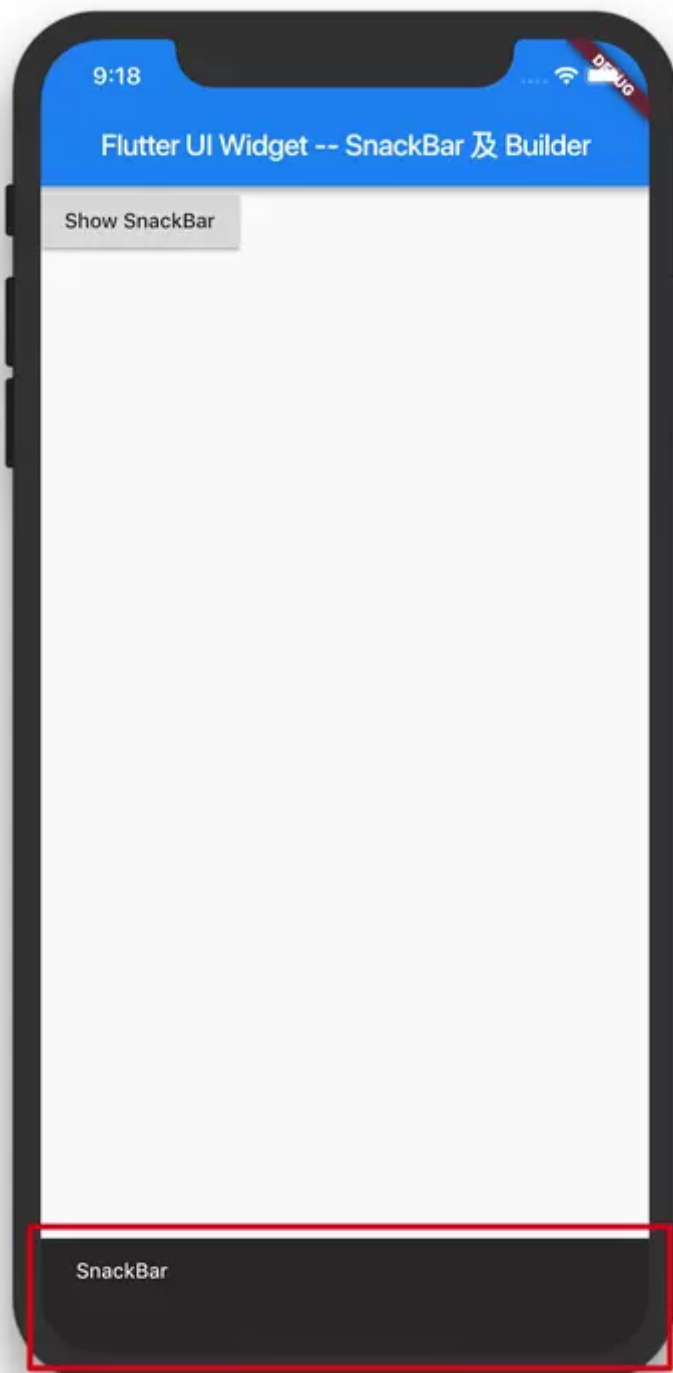
三、SnackBar分析

1、构造方法

2、其他参数

一、简介

SanckBar是Flutter的底部消息提示，在使用SnackBar的时候，需要使用Builder



二、使用

1、使用

```
1 Scaffold.of(context).showSnackBar(SnackBar(  
2   content: Text('SnackBar'), duration: Duration(seconds: 5)));
```

需要注意这里的context需要是Scaffold中的context，如果直接使用MaterialApp的context，则会报错

```
I/flutter (8385): EXCEPTION CAUGHT BY GESTURE  
I/flutter (8385): The following assertion was thrown while handling a gesture:  
I/flutter (8385): Scaffold.of() called with a context that does not contain a Scaffold.  
I/flutter (8385): No Scaffold ancestor could be found starting from the context that was passed to Scaffold.of(). This usually happens when the context provided is from the same StatefulWidget as that whose build function actually creates the Scaffold widget being sought.  
I/flutter (8385): There are several ways to avoid this problem. The simplest is to use a Builder to get a context that is "under" the Scaffold. For an example of this, please see the documentation for Scaffold.of():  
I/flutter (8385): https://docs.flutter.io/flutter/material/Scaffold/of.html  
I/flutter (8385): A more efficient solution is to split your build function into several widgets. This introduces a new context from which you can obtain the Scaffold. In this solution, you would have an outer widget that creates the Scaffold populated by instances of our new inner widgets, and then in these inner widgets you would use Scaffold.of().  
I/flutter (8385): A less elegant but more expedient solution is assign a GlobalKey to the Scaffold, then use the key.currentState property to obtain the ScaffoldState rather than using the Scaffold.of() function.
```

因此使用上面的方法需要两种方式:

- 使用BuildContext
- Snackbar中的Widget拆分出来

2、使用BuilderContext

```
1 class SnackBarBuilderWidget extends StatelessWidget {
2   @override
3   Widget build(BuildContext context) {
4     return MaterialApp(
5       title: "Flutter Demo",
6       theme: ThemeData(
7         primaryColor: Colors.blue,
8       ),
9       home: Scaffold(
10        appBar: AppBar(title: Text("Flutter UI Widget -- SnackBar 及 Builder")),

```

```

11  body: Builder(
12    builder: (context) => RaisedButton(
13      child: Text('Show SnackBar'),
14      onPressed: () {
15        Scaffold.of(context).showSnackBar(SnackBar(
16          content: Text('SnackBar'),
17          duration: Duration(seconds: 5)));
18      },
19    ),
20  ),
21  ),
22 );
23 }
24 }

```

3、将SnackBar的Widget拆分出来

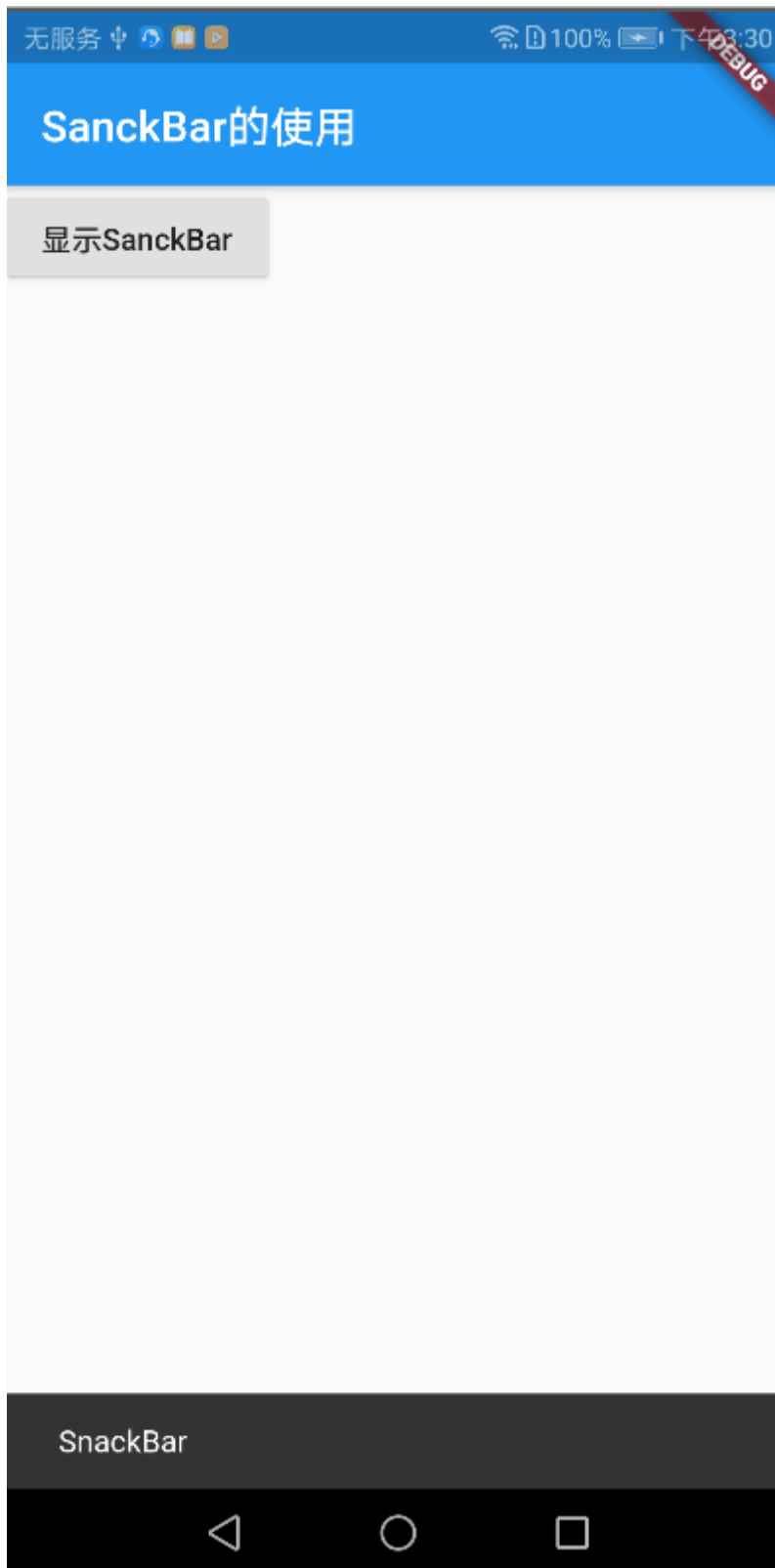
```

1  import 'package:flutter/cupertino.dart';
2  import 'package:flutter/material.dart';
3
4  void main() => runApp(SnackBarNoBuilerWidget());
5
6  class SnackBarNoBuilerWidget extends StatelessWidget {
7    @override
8    Widget build(BuildContext context) {
9      return MaterialApp(
10        title: "Flutter Demo",
11        theme: ThemeData(
12          primaryColor: Colors.blue,
13        ),
14        home: Scaffold(
15          appBar: AppBar(title: Text("Flutter UI基础Widget -- SnackBar")),
16          body: SnackBarWidget(),
17        );
18      }
19    }
20
21    class SnackBarWidget extends StatelessWidget {
22      @override
23      Widget build(BuildContext context) {
24        // TODO: implement build
25        return RaisedButton(

```

```
26   child: Text('Show SnackBar'),  
27   onPressed: () {  
28     Scaffold.of(context).showSnackBar(SnackBar(  
29       content: Text('SnackBar'), duration: Duration(seconds: 5)));  
30   },  
31   );  
32 }  
33 }
```

4、截图



三、SnackBar分析

1、构造方法

```
1 class SnackBar extends StatelessWidget {  
2   const SnackBar({  
3     Key key,
```

```
4  @required this.content,  
5  this.backgroundColor,  
6  this.action,  
7  this.duration = _kSnackBarDisplayDuration,  
8  this.animation,  
9  }) : assert(content != null),  
10  assert(duration != null),  
11  super(key: key);  
12  ...  
13 }
```

其中content和duration为必选参数

2、其他参数

参数名字	参数类型	意义	必选 or 可选
key	Key	Widget 的标识	可选
content	Widget	SnackBar 显示的主要内容	必选
backgroundColor	Color	SnackBar 的背景色	可选
action	SnackBarAction	SnackBar 的按钮	可选
duration	Duration	SnackBar 显示的时间 默认是4.0s	可选
animation	Animation	SnackBar 显示和消失的动画	可选