

# Flutter入门（13）：Flutter 组件之 Scaffold 详解



Maojunhao

关注

0.801

2020.09.11 11:15:04

字数 576

阅读 6,541

## 1. 基本介绍

Scaffold 提供了比较常见的页面属性。

Scaffold属性	介绍
appBar	页面上方导航条
body	页面容器
floatingActionButton	悬浮按钮
floatingActionButtonLocation	悬浮按钮位置
floatingActionButtonAnimator	悬浮按钮动画
persistentFooterButtons	显示在底部导航条上方的一组按钮
drawer	左侧菜单
endDrawer	右侧菜单
bottomNavigationBar	底部导航条
bottomSheet	一个持久停留在body下方，底部控件上方的控件
backgroundColor	背景色
resizeToAvoidBottomPadding	已废弃，resizeToAvoidBottomInset作为替代
resizeToAvoidBottomInset	默认为 true，防止一些小组件重复
primary	是否在屏幕顶部显示AppBar, 默认为 true，AppBar 是否向上延伸到状态栏，如电池电量，时间那一栏
drawerDragStartBehavior	控制 drawer 的一些特性
extendBody	body 是否延伸到底部控件
extendBodyBehindAppBar	默认 false，为 true 时，body 会置顶到 appBar 后，如appbar 为半透明色，可以有毛玻璃效果
drawerScrimColor	侧滑栏拉出来时，用来遮盖主页面的颜色
drawerEdgeDragWidth	侧滑栏拉出来的宽度
drawerEnableOpenDragGesture	左侧侧滑栏是否可以滑动
endDrawerEnableOpenDragGesture	右侧侧滑栏是否可以滑动

## 2. 示例代码

[代码下载地址](#)。如果对你有帮助的话记得给个关注，代码会根据[我的 Flutter 专题](#)不断更新。

## 3. Scaffold 组件

### 3.1 创建容器

优雅的编程，首先创建一个 scaffold.dart 文件。

```
1 import 'package:flutter/material.dart';
2
3 class FMScaffoldVC extends StatefulWidget {
4   FMScaffoldVC({Key key}) : super(key: key);
5   @override
6   FMScaffoldState createState() => FMScaffoldState();
7 }
8
9 class FMScaffoldState extends State<FMScaffoldVC> {
10  @override
11  Widget build(BuildContext context) {
12    // TODO: implement build
13    return Container(
14      child: Scaffold(
15        appBar: _appBar(),
16        body: _body(),
17      ),
18    );
19  }
20
21  AppBar _appBar(){
22    return AppBar(
23      title: Text('Scaffold'),
24      backgroundColor: Colors.lightBlue,
25    );
26  }
27
28  Container _body(){
29    return Container(
30      color: Colors.grey,
31    );
32  }
33 }
```

### 3.2 AppBar





scaffold appBar.png

### 3.3 floatingActionButton

我们给这个按钮增加一个返回事件，避免在使用其他属性时，导致页面缺失返回事件。

```
1  Widget build(BuildContext context) {  
2    // TODO: implement build  
3    return Container(  
4      child: Scaffold(  
5        appBar: _appBar(),  
6        body: _body(),  
7        floatingActionButton: _floatingActionButton(),  
8        floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,  
9        floatingActionButtonAnimator: FloatingActionButtonAnimator.scaling,  
10     ),  
11   );  
12 }  
13  
14 FloatingActionButton _floatingActionButton(){  
15   return FloatingActionButton(  
16     child: Text('返回'),  
17     onPressed: (){  
18       Navigator.pop(context);  
19     },  
20   );  
21 }
```

使用 `floatingActionButtonLocation` 可以改变按钮位置，可以自行尝试。



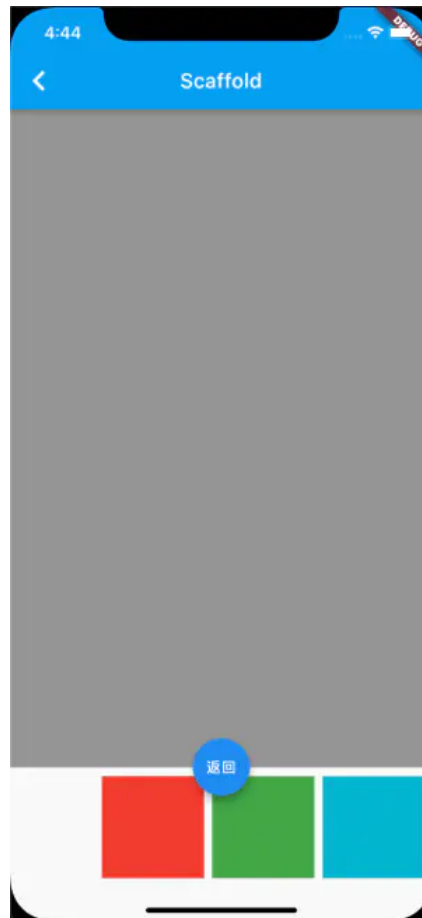
scaffold floatingActionButton.png

### 3.3 persistentFooterButtons

```

1      return Container(
2        child: Scaffold(
3          appBar: _appBar(),
4          body: _body(),
5          floatingActionButton: _floatingActionButton(),
6          floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,
7          floatingActionButtonAnimator: FloatingActionButtonAnimator.scaling,
8          persistentFooterButtons: _persistentFooterButtons(),
9        ),
10     );
11   }
12
13   List<Widget> _persistentFooterButtons(){
14     return [
15       Container(
16         width: 100,
17         height: 100,
18         color: Colors.red,
19       ),
20       Container(
21         width: 100,
22         height: 100,
23         color: Colors.green,
24       ),
25       Container(
26         width: 100,
27         height: 100,
28         color: Colors.cyan,
29       ),
30     ];
31   }

```



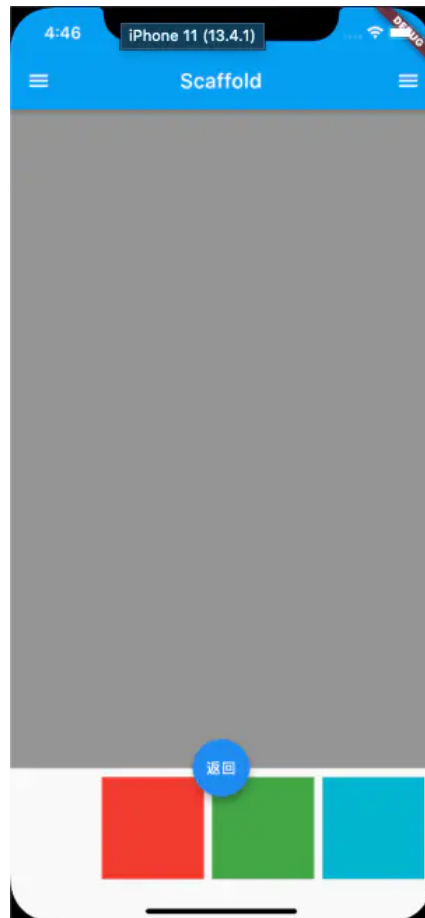
scaffold persistentFooterButtons.png

### 3.4 drawer / endDrawer

drawer / endDrawer 可以通过点击左上角，右上角按钮触发，也可以左滑，右滑触发。  
 drawerEnableOpenDragGesture 默认为 true，设置 drawer 是否右滑触发  
 endDrawerEnableOpenDragGesture 默认为 true，设置 endDrawer 是否左滑触发



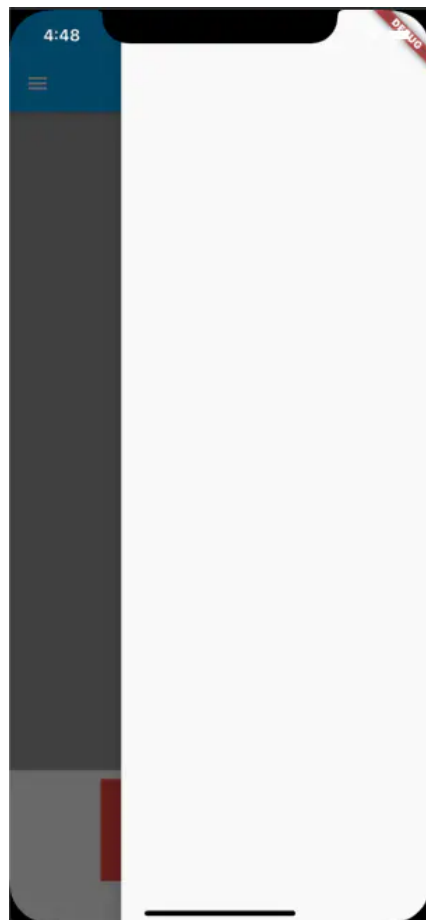
```
1 Widget build(BuildContext context) {  
2   // TODO: implement build  
3   return Container(  
4     child: Scaffold(  
5       appBar: _appBar(),  
6       body: _body(),  
7       floatingActionButton: _floatingActionButton(),  
8       floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,  
9       floatingActionButtonAnimator: FloatingActionButtonAnimator.scaling,  
10      persistentFooterButtons: _persistentFooterButtons(),  
11      drawer: Drawer(),  
12      endDrawer: Drawer(),  
13    ),  
14  );  
15 }
```



scaffold drawer home.png



scaffold drawer.png



scaffold enddrawer.png

### 3.5 bottomNavigationBar

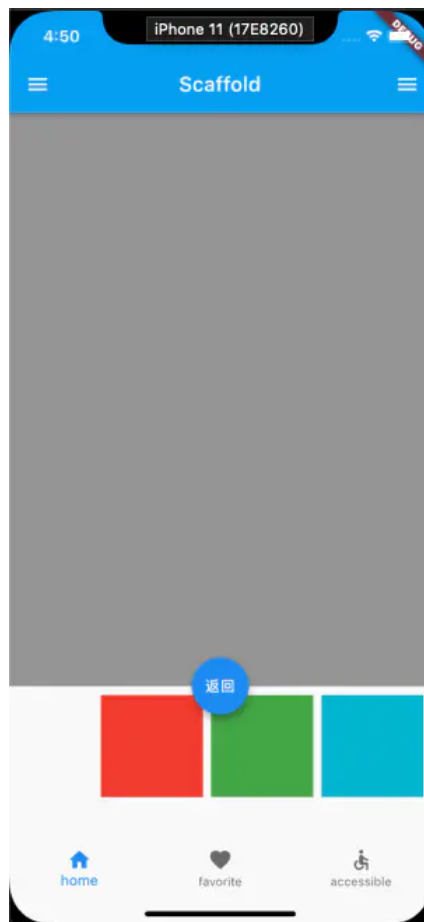
```
1  Widget build(BuildContext context) {
2    // TODO: implement build
3    return Container(
4      child: Scaffold(
5        appBar: _appBar(),
6        body: _body(),
7        floatingActionButton: _floatingActionButton(),
8        floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,
9        floatingActionButtonAnimator: FloatingActionButtonAnimator.scaling,
10       persistentFooterButtons: _persistentFooterButtons(),
11       drawer: Drawer(),
12       endDrawer: Drawer(),
13       bottomNavigationBar: _bottomNavigationBar(),
14     ),
15   );
16 }
17
18 BottomNavigationBar _bottomNavigationBar(){
19   return BottomNavigationBar(
20     items: [
21       BottomNavigationBarItem(
```



```

22         icon: Icon(
23             Icons.home
24         ),
25         title: Text('home'),
26     ),
27     BottomNavigationBarItem(
28         icon: Icon(
29             Icons.favorite
30         ),
31         title: Text('favorite'),
32     ),
33     BottomNavigationBarItem(
34         icon: Icon(
35             Icons.accessible
36         ),
37         title: Text('accessible'),
38     ),
39 ],
40 );
41 }

```



scaffold bottomNavigationBar.png

### 3.6 bottomSheet

```

1  Widget build(BuildContext context) {
2      // TODO: implement build
3      return Container(
4          child: Scaffold(
5              appBar: _appBar(),
6              body: _body(),
7              floatingActionButton: _floatingActionButton(),
8              floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,
9              floatingActionButtonAnimator: FloatingActionButtonAnimator.scaling,
10             persistentFooterButtons: _persistentFooterButtons(),
11             drawer: Drawer(),
12             endDrawer: Drawer(),
13             bottomNavigationBar: _bottomNavigationBar(),
14             bottomSheet: _bottomSheet(),
15         ),
16     );
17 }

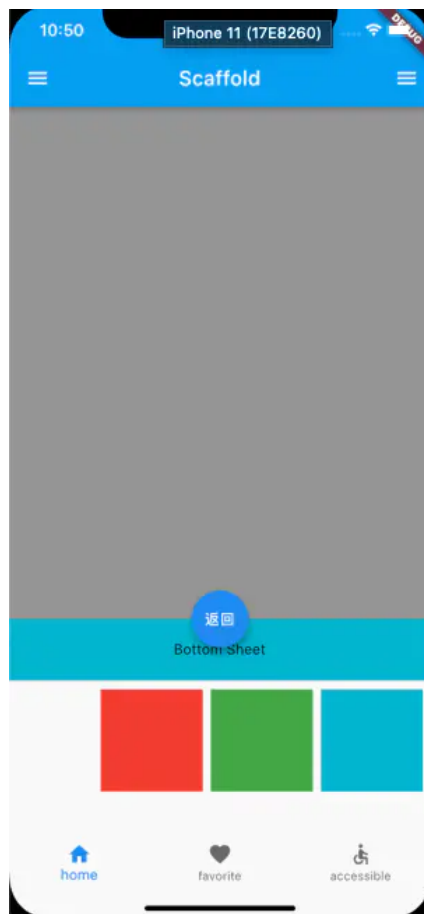
```



```

18
19 BottomSheet _bottomSheet(){
20   return BottomSheet(
21     onClosing: (){},
22     builder: (BuildContext context){
23       return Container(
24         height: 60,
25         color: Colors.cyan,
26         child: Text('Bottom Sheet'),
27         alignment: Alignment.center,
28       );
29     },
30   );

```



scaffold bottomSheet.png

### 3.7 backgroundColor

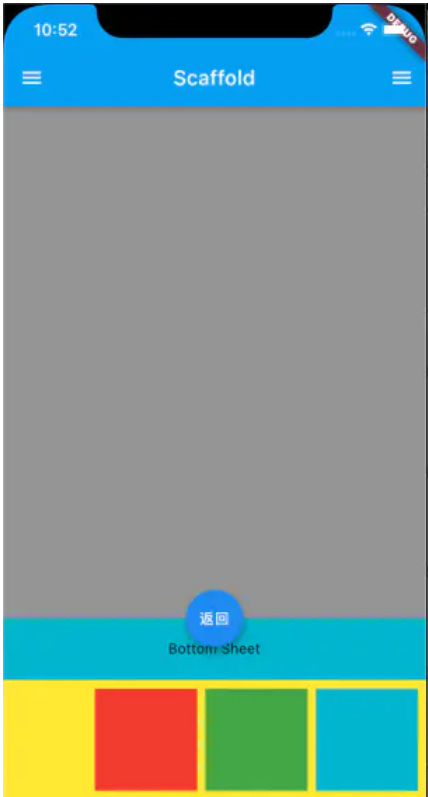
```

1  Widget build(BuildContext context) {
2    // TODO: implement build
3    return Container(
4      child: Scaffold(
5        appBar: _appBar(),
6        body: _body(),
7        floatingActionButton: _floatingActionButton(),
8        floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,
9        floatingActionButtonAnimator: FloatingActionButtonAnimator.scaling,
10       persistentFooterButtons: _persistentFooterButtons(),
11       drawer: Drawer(),
12       endDrawer: Drawer(),
13       bottomNavigationBar: _bottomNavigationBar(),
14       bottomSheet: _bottomSheet(),
15       backgroundColor: Colors.yellow,
16     ),
17   );
18 }

```







scaffold backgroundColor.png

### 3.8 其他属性


还有一些 bool 值属性，用的场景较少，有需要的自行了解。


### 4. 技术小结

- 了解 Scaffold 提供了哪些控件。
- Scaffold 属性介绍。
- 基础 Scaffold 控件的使用。
- 基础 Scaffold 的布局效果。

 8人点赞 >



 Flutter实战笔记





Maojunhao 打工人  
总资产101 共写了4.1W字 获得216个赞 共196个粉丝

关注

