

## Scaffold类

实现了基本的材料设计可视化布局结构。

这个类提供了api让抽屉,小吃店,下表。

显示一间 的控件,获得ScaffoldState对当前BuildContext通过Scaffold.of并使用ScaffoldState.showSnackBar和ScaffoldState.showBottomSheet功能。

参见:

- [AppBar](#)通常,这是一个单杠显示应用程序的顶部 使用AppBar财产。
- [BottomAppBar](#)通常,这是一个单杠底部所示 一个应用程序使用bottomNavigationBar财产。
- [FloatingActionButton](#),这是一个圆形按钮通常所示 右下角的应用程序使用floatingActionButton财产。
- [FloatingActionButtonLocation](#),用于放置floatingActionButton在脚手架的布局。
- [FloatingActionButtonAnimator](#)用于动画floatingActionButton从一个floatingActionButtonLocation来 另一个地方。
- [抽屉里](#),这是一个垂直面板通常是显示的 离开身体的(而且往往隐藏在手机)使用抽屉里财产。
- [BottomNavigationBar](#)通常,这是一个水平的按钮 底部显示的应用程序使用bottomNavigationBar财产。
- [间小吃店](#)通常,这是一个临时通知附近的显示 的应用程序使用ScaffoldState.showSnackBar方法。
- [BottomSheet](#)通常,这是一个叠加显示的底部附近 应用。下表可以是持续的,在这种情况下,它显示 使用ScaffoldState.showBottomSheet方法,或模态,在这种情况下 它显示了使用showModalBottomSheet函数。
- [ScaffoldState](#),是与这个小部件相关联的状态。
- [material.google.com/layout/structure.html](https://material.google.com/layout/structure.html)

继承

- [对象](#)
- [Diagnosticable](#)
- [DiagnosticableTree](#)
- [小部件](#)
- [StatefulWidget](#)
- 脚手架

## 构造函数

[脚手架](#)({[关键](#) 关键, [PreferredSizeWidget](#) appBar, [小部件](#) 身体, [小部件](#) floatingActionButton, [FloatingActionButtonLocation](#) floatingActionButtonLocation, [FloatingActionButtonAnimator](#) 列表<[小部件](#)> persistentFooterButtons, [小部件](#) 抽屉里, [小部件](#) endDrawer, [小部件](#) bottomNavigationBar, [颜色](#) 写成backgroundColor, [bool](#) resizeToAvoidBottomPadding:真正的, [bool](#) 主:真正的})

创建一个视觉脚手架材料设计小部件。

常量

## 属性

[AppBar](#) → [PreferredSizeWidget](#)

应用程序栏显示的顶部支架。

最后

[写成backgroundColor](#) → [颜色](#)

的颜色Material整个脚手架部件构成。 [\[...\]](#)

最后

[身体](#) → [小部件](#)

脚手架的主要内容。 [\[...\]](#)

最后

[bottomNavigationBar](#) → [小部件](#)

底部导航栏显示底部的支架。 [\[...\]](#)

最后

[抽屉里](#) → [小部件](#)

一个面板显示的body, 往往隐藏在移动 设备。 刷的从左到右(TextDirection.ltr)或 从右到左(TextDirection.rtl) [\[...\]](#)

最后

[endDrawer](#) → [小部件](#)

一个面板显示的body, 往往隐藏在移动 设备。 刷的从右到左(TextDirection.ltr)或 从左到右(TextDirection.rtl) [\[...\]](#)

最后

[floatingActionButton](#) → [小部件](#)

一个按钮显示上面浮动body在右下角。 [\[...\]](#)

最后

[floatingActionButtonAnimator](#) → [FloatingActionButtonAnimator](#)

动画师将floatingActionButton到一个新的floatingActionButtonLocation。 [\[...\]](#)

最后

[floatingActionButtonLocation](#) → [FloatingActionButtonLocation](#)

负责确定的地方floatingActionButton应该去。 [\[...\]](#)

最后

[persistentFooterButtons](#) → [列表<小部件>](#)

一组显示的按钮底部的支架。 [\[...\]](#)

最后

[主](#) → [bool](#)

这个脚手架是否显示在屏幕的顶部。 [\[...\]](#)

最后

[resizeToAvoidBottomPadding](#) → [bool](#)

是否body(和其他浮动窗口小部件)应大小自己 避免窗口的底部填充。 [\[...\]](#)

最后

[hashCode](#) → [int](#)

这个对象的哈希码。 [\[...\]](#)

只读的, 遗传的

[关键](#) → [关键](#)

控制一个小部件替换另一个小部件在树上。 [\[...\]](#)

最后, 继承了

[runtimeType](#) → [类型](#)

一个对象的运行时类型的代表。

只读的, 遗传的

## 方法

[createState\(\)](#) → [ScaffoldState](#)

为这个小部件创建可变状态给定树中的位置。 [\[...\]](#)

[`createElement\(\)`](#) → [`StatefulElement`](#)

创建一个[`StatefulElement`](#)在树上来管理这个小部件的位置。[...]

继承了

[`debugDescribeChildren\(\)`](#) → [列表<`DiagnosticsNode`>](#)

返回一个列表[`DiagnosticsNode`](#)描述该节点的对象 的孩子。[...]

@protected, 继承了

[`debugFillProperties`](#)([`DiagnosticPropertiesBuilder`](#) 属性) → 无效

添加额外的属性与节点相关联。[...]

继承了

[`noSuchMethod`](#)([调用](#) 调用) → 动态

当用户访问一个不存在的方法或属性调用。[...]

继承了

[`toDiagnosticsNode`](#)([{字符串 的名字, `DiagnosticsTreeStyle` 风格}](#)) → [`DiagnosticsNode`](#)

返回一个对象被调试的调试表示 工具和[`toStringDeep`](#)。[...]

继承了

[`toString`](#)([`DiagnosticLevel`](#) minLevel:[`DiagnosticLevel.debug`](#)) → [字符串](#)

返回该对象的字符串表示。

继承了

[`toStringDeep`](#)([{字符串 prefixLineOne:"](#), [字符](#)

[串 prefixOtherLines, `DiagnosticLevel`](#) minLevel:[`DiagnosticLevel.debug`](#)) → [字符串](#)

返回一个字符串表示该节点及其后代。[...]

继承了

[`toStringShallow`](#)([{字符串 乔伊纳:"](#)、[“](#), [`DiagnosticLevel`](#) minLevel:[`DiagnosticLevel.debug`](#)) → [字符串](#)

返回一行详细描述的对象。[...]

继承了

[`toStringShort`](#)() → [字符串](#)

短, 这个小部件的文本描述。

继承了

## 操作

[`运算符=`](#)([动态](#) 其他) → [bool](#)

相等操作符。[...]

继承了

## 静态方法

[`geometryOf`](#)([`BuildContext`](#) 上下文) → [`ValueListenable<ScaffoldGeometry>`](#)

返回一个[`ValueListenable`](#)为[`ScaffoldGeometry`](#)为最接近的[脚手架](#)的祖先给定的上下文。[...]

[`hasDrawer`](#)([`BuildContext`](#) 上下文, { [bool](#) registerForUpdates:真正的}) → [bool](#)

是否包含最严密的脚手架有一个给定的上下文 抽屉里。[...]

[的](#)([`BuildContext`](#) 上下文, { [bool](#) nullOk:假}) → [`ScaffoldState`](#)

最近这类的实例的状态包含给定的上下文。[...]

[`setFloatingActionButtonNotchFor`](#)([`BuildContext`](#) 上下文, [`ComputeNotch`](#) computeNotch) → [`VoidCallback`](#)

设置[`ScaffoldGeometry.floatingActionButtonNotch`](#)为最接近的[脚手架](#)给定上下文的祖先, 如果一个人的存在。[...]