

一、状态管理、Flutter Getx介绍

1.1、状态管理

通俗的讲：当我们想在多个页面（组件/Widget）之间共享状态（数据），或者一个页面（组件/Widget）中的多个子组件之间共享状态（数据），这个时候我们就可以用Flutter中的状态管理来管理统一的状态（数据），实现不同组件之间的传值和数据共享。

现在Flutter的状态管理方案很多，redux、bloc、state、provider、Getx。

provider是官方提供的状态管理解决方案，主要功能就是状态管理。Getx是第三方的状态管理插件，不仅具有状态管理的功能，还具有路由管理、主题管理、国际化多语言管理、Obx局部更新、网络请求、数据验证等功能，相比其他状态管理插件Getx 简单、功能强大并且高性能。

1.2、Flutter Getx介绍

GetX 是 Flutter 上的一个轻量且强大的解决方案，GetX为我们提供了高性能的状态管理、智能的依赖注入和便捷的路由管理。

- GetX 有3个基本原则：
 - **性能：** GetX 专注于性能和最小资源消耗。GetX 打包后的apk占用大小和运行时的内存占用与其他状态管理插件不相上下。
 - **效率：** GetX 的语法非常简捷，并保持了极高的性能，能极大缩短你的开发时长。
 - **结构：** GetX 可以将界面、逻辑、依赖和路由完全解耦，用起来更清爽，逻辑更清晰，代码更容易维护
- **GetX 并不臃肿，却很轻量。如果你只使用状态管理，只有状态管理模块会被编译，其他没用到的东西都不会被编译到你的代码中。它拥有众多的功能，但这些功能都在独立的容器中，只有在使用后才会启动。**
- Getx有一个庞大的生态系统，能够在Android、iOS、Web、Mac、Linux、Windows和你的服务器上运行。通过[Get Server](#)可以在你的后端完全重用你在前端写的代码。

官网：<https://pub.dev/packages/get>

中文文档：<https://github.com/jonataslaw/getx/blob/master/README.zh-cn.md>

二、Flutter Getx 中的Dialog 以及改变主题

2.1 Getx安装

将 Get 添加到你的 pubspec.yaml 文件中。

```
dependencies:  
  get: ^4.6.5
```

在需要用到的文件中导入，它将被使用。

```
import 'package:get/get.dart';
```

2.2 Getx 使用 Dialog

一、设置应用程序入口

当我们导入依赖后，在应用程序顶层把 `GetMaterialApp` 作为顶层，如下所示

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  widget build(BuildContext context) {
    return GetMaterialApp(
      title: "GetX",
      home: Scaffold(
        appBar: AppBar(title: Text("GetX Title")),
      ),
    );
  }
}
```

二、调用BottomSheet 以及改变主题

我们可以通过 `GetX` 很轻松的调用 `bottomSheet()`，而且无需传入 `context`，下面我给出一个例子，使用 `GetX` 弹出 `bottomSheet` 并很轻松的实现切换主题。

我们可以通过 `Get.bottomSheet()` 来显示 `BottomSheet`，通过 `Get.back()` 实现路由返回，通过 `Get.changeTheme(ThemeData.dark())` 切换皮肤主题，通过 `Get.isDarkMode` 判断主题样式。

```
ElevatedButton(
  onPressed: () {
    Get.bottomSheet(Container(
      color: Get.isDarkMode ? Colors.black12 : Colors.white,
      height: 200,
      child: Column(
        children: [
          ListTile(
            leading: Icon(Icons.wb_sunny_outlined,
              color:
                Get.isDarkMode ? Colors.white : Colors.black),
            title: Text("白天模式",
              style: TextStyle(
                color: Get.isDarkMode
                  ? Colors.white
                  : Colors.black)),
            onTap: () {
```

```
        Get.changeTheme(ThemeData.light());
        Get.back();
    },
),
ListTile(
  leading: Icon(Icons.wb_sunny,
    color:
      Get.isDarkMode ? Colors.white : Colors.black),
  title: Text("黑夜模式",
    style: TextStyle(
      color: Get.isDarkMode
        ? Colors.white
        : Colors.black)),
  onTap: () {
    Get.changeTheme(ThemeData.dark());
    Get.back();
  },
),
],
),
));
},
child: const Text("Show BottomSheet"))
```

BottomSheet属性和说明

字段	属性	描述
bottomsheet	Widget	弹出的Widget组件
backgroundColor	Color	bottomsheet的背景颜色
elevation	double	bottomsheet的阴影
persistent	bool	是否添加到路由中
shape	ShapeBorder	边框形状，一般用于圆角效果
clipBehavior	Clip	裁剪的方式
barrierColor	Color	弹出层的背景颜色
ignoreSafeArea	bool	是否忽略安全适配
isScrollControlled	bool	是否支持全屏弹出，默认false
useRootNavigator	bool	是否使用根导航
isDismissible	bool	点击背景是否可关闭，默认ture
enableDrag	bool	是否可以拖动关闭，默认true
settings	RouteSettings	路由设置
enterBottomSheetDuration	Duration	bottomsheet进入时的动画时间

字段	属性	描述
exitBottomSheetDuration	Duration	bottomsheet退出时的动画时间

三、调用snackbar

Snackbar 和我们前面讲的toast有点相似， 如果想在应用程序中触发某些特定的事件后， 需要弹出快捷消息， 那么使用 Snackbar 则是最佳的选择。

我们可以通过 Get.snackbar() 来显示 snackbar ， 如下所示

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  widget build(BuildContext context) {
    return GetMaterialApp(
      title: "GetX",
      home: Scaffold(
        appBar: AppBar(
          title: Text("GetX Title"),
        ),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.center,
            children: [
              ElevatedButton(
                onPressed: () {
                  Get.snackbar("Snackbar 标题", "欢迎使用Snackbar");
                },
                child: Text("显示 Snackbar")
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

如果您运行了代码，那么恭喜你，你已经会用 Getx 来展示 snackbar 了。你将得到下面的结果

Snackbar属性和说明

字段	属性	描述
title	String	弹出的标题文字
message	String	弹出的消息文字
colorText	Color	title和message的文字颜色
duration	Duration	Snackbar弹出的持续时间（默认3秒）
instantInit	bool	当false可以把snackbar 放在 initState，默认true
snackPosition	SnackPosition	弹出时的位置，有两个选项【TOP，BOTTOM】默认TOP
titleText	Widget	弹出标题的组件，设置该属性会导致 title属性失效
messageText	Widget	弹出消息的组件，设置该属性会导致 messageText属性失效
icon	Widget	弹出时图标，显示在title和message的左侧
shouldIconPulse	bool	弹出时图标是否闪烁，默认false
maxWidth	double	Snackbar最大的宽度
margin	EdgeInsets	Snackbar外边距，默认zero
padding	EdgeInsets	Snackbar内边距，默认 EdgeInsets.all(16)
borderRadius	double	边框圆角大小，默认15
borderColor	Color	边框的颜色，必须设置borderWidth，否则无效果
borderWidth	double	边框的线条宽度
backgroundColor	Color	Snackbar背景颜色，默认 Colors.grey.withOpacity(0.2)
leftBarIndicatorColor	Color	左侧指示器的颜色
boxShadows	List	Snackbar阴影颜色
backgroundGradient	Gradient	背景的线性颜色
mainButton	TextButton	主要按钮，一般显示发送、确认按钮
onTap	OnTap	点击Snackbar事件回调
isDismissible	bool	是否开启Snackbar手势关闭，可配合 dismissDirection使用
showProgressIndicator	bool	是否显示进度条指示器，默认false
dismissDirection	SnackDismissDirection	Snackbar关闭的方向
progressIndicatorController	AnimationController	进度条指示器的动画控制器
progressIndicatorBackgroundColor	Color	进度条指示器的背景颜色
progressIndicatorValueColor	Animation	进度条指示器的背景颜色，Animation
snackStyle	SnackStyle	Snackbar是否会附加到屏幕边缘