

Flutter 中的普通路由、普通路由传值、命名路由、命名路由传值

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html> （IT 营）

一、Flutter 中的普通路由.....	1
二、Flutter 中的基本路由使用.....	1
三、Flutter 中的基本路由跳转传值.....	2
四、Flutter 中的命名路由.....	3
五、Flutter 中的命名路由跳转传值.....	4
六、Vscode 调试 Flutter 项目.....	6

一、Flutter 中的路由

Flutter 中的路由通俗的讲就是页面跳转。在 Flutter 中通过 Navigator 组件管理路由导航。并提供了管理堆栈的方法。如：Navigator.push 和 Navigator.pop

Flutter 中给我们提供了两种配置路由跳转的方式：1、基本路由 2、命名路由

二、Flutter 中的基本路由使用

比如我们现在想从 HomePage 组件跳转到 SearchPage 组件。

1、需要在 HomePage 中引入 SearchPage.dart

```
import '../SearchPage.dart';
```

2、在 HomePage 中通过下面方法跳转

```
RaisedButton(  
  child: Text("跳转到搜索页面"),  
  onPressed: (){  
    Navigator.of(context).push(  
      MaterialPageRoute(  
        builder: (BuildContext context){  
          return SearchPage();  
        }  
      )  
    );  
  },  
  color: Theme.of(context).accentColor,  
  textTheme: ButtonTextTheme.primary  
)
```

三、Flutter 中的基本路由跳转传值

比如我们现在想从 HomePage 组件跳转到 SearchPage 组件传值。

1、需要在 HomePage 中引入 SearchPage.dart

```
import '../SearchPage.dart';
```

2、在 HomePage 中通过下面方法跳转

```
RaisedButton(  
  child: Text("跳转到搜索页面"),  
  onPressed: (){  
    Navigator.of(context).push(  
      MaterialPageRoute(  
        builder: (BuildContext context){  
          return SearchPage(title:"表单"); //传值  
        }  
      )  
    );  
  },  
  color: Theme.of(context).accentColor,
```

```
        textTheme: ButtonTextTheme.primary  
    )
```

四、Flutter 中的命名路由

1、配置路由

```
import 'package:flutter/material.dart';  
  
import 'pages/Tabs.dart';  
  
import 'pages/Search.dart';  
import 'pages/Form.dart';  
  
void main() => runApp(MyApp());  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      // home: Tabs(),  
      initialRoute: '/',  
      routes: {  
        '/': (context) => Tabs(),  
        '/search': (context) => SearchPage(),  
        '/form': (context) => FormPage(),  
      },  
    );  
  }  
}
```

2、路由跳转

```
RaisedButton(  
  child: Text("跳转到搜索页面"),  
  onPressed: () {  
    Navigator.pushNamed(context, '/search');
```

```
    },  
    color: Theme.of(context).accentColor,  
    textTheme: ButtonTextTheme.primary  
  )
```

五、Flutter 中的命名路由跳转传值

官方文档: <https://flutter.dev/docs/cookbook/navigation/navigate-with-arguments>

1、配置路由:

```
import 'package:flutter/material.dart';  
import 'pages/Tabs.dart';  
import 'pages/Search.dart';  
import 'pages/Form.dart';  
  
void main() => runApp(MyApp());  
  
class MyApp extends StatelessWidget {  
  final Map<String,Function> routes = {  
    '/': (context) => Tabs(),  
    '/search': (context) => SearchPage(),  
    '/form': (context, {arguments}) => FormPage(arguments: arguments),  
  };  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: Tabs(),  
      onGenerateRoute: (RouteSettings settings) {  
        // 统一处理  
        final String? name = settings.name;  
        final Function? pageContentBuilder = this.routes[name];  
        if (pageContentBuilder != null) {  
          if (settings.arguments != null) {  
            final Route route = MaterialPageRoute(  
              builder: (context) => pageContentBuilder(context,  
                arguments: settings.arguments));  
            return route;  
          }  
        }  
      }  
    );  
  }  
}
```

```
        } else {  
            final Route route = MaterialPageRoute(  
                builder: (context) => pageContentBuilder(context));  
            return route;  
        }  
    }  
});  
}  
}
```

2、跳转传值

```
RaisedButton(  
    child: Text("跳转到表单演示页面"),  
    onPressed: () {  
        Navigator.pushNamed(context, '/form', arguments: {  
            "id": 20  
        });  
    },  
    color: Theme.of(context).accentColor,  
    textTheme: ButtonTextTheme.primary  
)
```

3、接收参数

```
import 'package:flutter/material.dart';  
  
class FormPage extends StatelessWidget {  
    final Map arguments;  
    FormPage({this.arguments});  
  
    @override  
    Widget build(BuildContext context) {  
        return Scaffold(  
            appBar: AppBar(  
                title: Text("搜索"),  
            ),  
            body: Text("我是一个表单页面 ${arguments != null ? arguments['id'] : '0'}")  
        );  
    }  
}
```

```
}  
}
```

六、Flutter 中的命名路单独抽离到一个文件

```
import 'package:flutter/material.dart';  
  
import '../pages/Tabs.dart';  
import '../pages/Form.dart';  
import '../pages/Search.dart';  
import '../pages/Product.dart';  
import '../pages/ProductInfo.dart';  
  
//配置路由,定义 Map 类型的 routes,Key 为 String 类型, Value 为 Function 类型  
final Map<String,Function> routes={  
  '/:(context)=>Tabs(),  
  '/form':(context)=>FormPage(),  
  '/product':(context)=>ProductPage(),  
  '/productinfo':(context,{arguments})=>ProductInfoPage(arguments:arguments),  
  '/search':(context,{arguments})=>SearchPage(arguments:arguments),  
};  
  
//固定写法  
var onGenerateRoute=(RouteSettings settings) {  
  //String? 表示 name 为可空类型  
  final String? name = settings.name;  
  //Function? 表示 pageContentBuilder 为可空类型  
  final Function? pageContentBuilder = routes[name];  
  if (pageContentBuilder != null) {  
    if (settings.arguments != null) {  
      final Route route = MaterialPageRoute(  
        builder: (context) =>  
          pageContentBuilder(context, arguments: settings.arguments));  
      return route;  
    }else{  
      final Route route = MaterialPageRoute(  
        builder: (context) =>
```

```
        pageContentBuilder(context));  
        return route;  
    }  
}  
};
```

```
import 'package:flutter/material.dart';  
import 'routes/Routes.dart';  
void main() => runApp(MyApp());  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      // home: Tabs(),  
      initialRoute: '/',  
      onGenerateRoute: onGenerateRoute  
    );  
  }  
}
```

七、Vscode 调试 Flutter 项目

1、Vscode 中打开 flutter 项目进行开发

2、运行 Flutter 项目

```
flutter run
```

r 键： 点击后热加载，也就算是重新加载吧。

p 键： 显示网格，这个可以很好的掌握布局情况，工作中很有用。

o 键： 切换 android 和 ios 的预览模式。

q 键： 退出调试预览模式。

3、Vscode 默认连不上第三方模拟器解决方案

cd 到对应夜神模拟器 D:\Program Files\Nox\bin 目录 然后运行

```
nox_adb.exe connect 127.0.0.1:62001
```