

路由与数据传递

2021年7月2日 9:49

在flutter中路由用Route类来进行表示，Navigator是对Route进行管理的Widget。

路由表的格式为Map<String, WidgetBuilder>，key为路由名称，value是回调函数，用于生产相应的路由Widget。

一、路由间数据传递

1.1 页面跳转数据传递

一个页面跳转到一个新的页面数据传递的方式主要有以下两种：

1. 通过Navigator.push()或者Navigator.pushNamed()方式传递数据
2. 通过Widget的构造函数传递数据

```
class CustomArguments {
  String content;
  CustomArguments(this.content);
}

class FirstPage extends StatelessWidget {
  const FirstPage({Key? key}): super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('第一页'),
      ),
      body: Center(
        child: ElevatedButton(
          child: Text('跳转到第二页'),
          onPressed: () {
            Navigator.pushNamed(context, '/Second', arguments: CustomArguments("helloworld"));
          },
        ),
      ),
    );
  }
}

class SecondPage extends StatelessWidget {
  const SecondPage({Key? key}): super(key: key);
  @override
  Widget build(BuildContext context) {
    Object? object = ModalRoute.of(context)!.settings.arguments;
    CustomArguments customArguments;
    if (object == null) {
      customArguments = new CustomArguments("error");
    } else {
      customArguments = object as CustomArguments;
    }
  }
}
```

```
return Scaffold(  
  appBar: AppBar(  
    title: Text('第二页'),  
  ),  
  body: Center(  
    child: Column(  
      children: <Widget>[  
        Text('第一页的数据为: '),  
        Text(customArguments.content),  
      ],  
    ),  
  ),  
);  
}
```

在第一个界面中pushNamed方法增加了参数arguments，第二个界面中ModalRoute.of(context)!.settings.arguments获取参数。

3.2 页面跳转返回数据

第一页使用如下方法跳转

```
_navigateToSecondPage(BuildContext context) async {  
  dynamic customArguments = await Navigator.pushNamed(context, '/Second', arguments: CustomArguments("第一页到第二页的数据"));  
  print(customArguments.content);  
}
```

即异步获取数据

第二界面在pop中增加数据参数即可。