Flutter之shared_preferences的使用、源码分析(一)

卢叁

`shared_preferences`介绍

`shared_preferences`主要的作用是用于将数据异步持久化到磁盘，因为持久化数据只是存储到临时目录，当app删除时该存储的数据就是消失，web开发时清除浏览器存储的数据也将消失。

支持存储类型：

1.bool

2.int

3.double

4.string

5.stringList

`shared_preferences`应用场景
主要用于持久化数据，如持久化用户信息、列表数据等。

持久化用户信息

因为用户信息基本是不改变的，而在一个应用程序中常常会有多个页面需要展示用户信息，我们不可能每次都去获取接口，那么本地持久化就会变得很方便。

持久化列表数据

为了给用户更好的体验，在获取列表数据时我们常常会先展示旧数据，带给用户更好的体验，不至于一打开页面就是空白的，当我们采用持久化列表数据后，可以直接先展示本地数据，当网络数据请求回来后在进行数据更新。

`shared_preferences`使用的对应类库

我们知道每个平台持久化数据的方式都不一样，而shared_preferences针对不同的平台封装了一个通用的类库，接下来我们看看不同平台下他们使用的库

iOS： NSUserDefaults
Android： SharedPreferences
Web： localStorage
Linux： FileSystem（保存数据到本地系统文件库中）
Mac OS： FileSystem（保存数据到本地系统文件库中）
Windows： FileSystem（保存数据到本地系统文件库中）

`shared_preferences`基本使用

`pubspec.yaml`导入依赖

```
shared_preferences: ^2.0.8
```

然后cd 到工程目录下执行flutter packages get命令，获取依赖包

导入头文件

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

获取实例对象

```
SharedPreferences? sharedPreferences = await
SharedPreferences.getInstance();
```

设置持久化数据

我们可以通过`sharedPreferences`的实例化对象调用对应的`set`方法设置持久化数据

```
// 设置持久化数据
void _setData() async {
  // 实例化
  sharedPreferences = await
SharedPreferences.getInstance();

  // 设置string类型
  await sharedPreferences?.setString("name", "Jimi");

  // 设置int类型
  await sharedPreferences?.setInt("age", 18);

  // 设置bool类型
  await sharedPreferences?.setBool("isTeacher", true);

  // 设置double类型
  await sharedPreferences?.setDouble("height", 1.88);
```

```
 // 设置string类型的数组
  await sharedPreferences?.setStringList("action", ["吃饭",
"睡觉", "打豆豆"]);

  setState(() {});
}
```

读取持久化数据

我们可以通过`sharedPreferences`的实例化对象调用对应的`get`方法读取持久化数据

```
Text("名字: ${sharedPreferences?.getString("name") ?? ""}",
     style: TextStyle(
       color: Colors.blue,
       fontSize: 20
     ),
    ),
SizedBox(height: 20,),
Text("年龄: ${sharedPreferences?.getInt("age") ?? ""}",
     style: TextStyle(
       color: Colors.red,
       fontSize: 20
     ),
    ),
SizedBox(height: 20,),
Text("是老师吗?: ${sharedPreferences?.getBool("isTeacher") ??
""}",
     style: TextStyle(
       color: Colors.orange,
       fontSize: 20
     ),
    ),
SizedBox(height: 20,),
```

```
Text("身高: ${sharedPreferences?.getDouble("height") ??
""}",
     style: TextStyle(
       color: Colors.pink,
       fontSize: 20
     ),
    ),
SizedBox(height: 20,),
Text("我正在: $
{sharedPreferences?.getStringList("action") ?? ""}",
     style: TextStyle(
       color: Colors.purple,
       fontSize: 20
     ),
    ),
```

完整代码

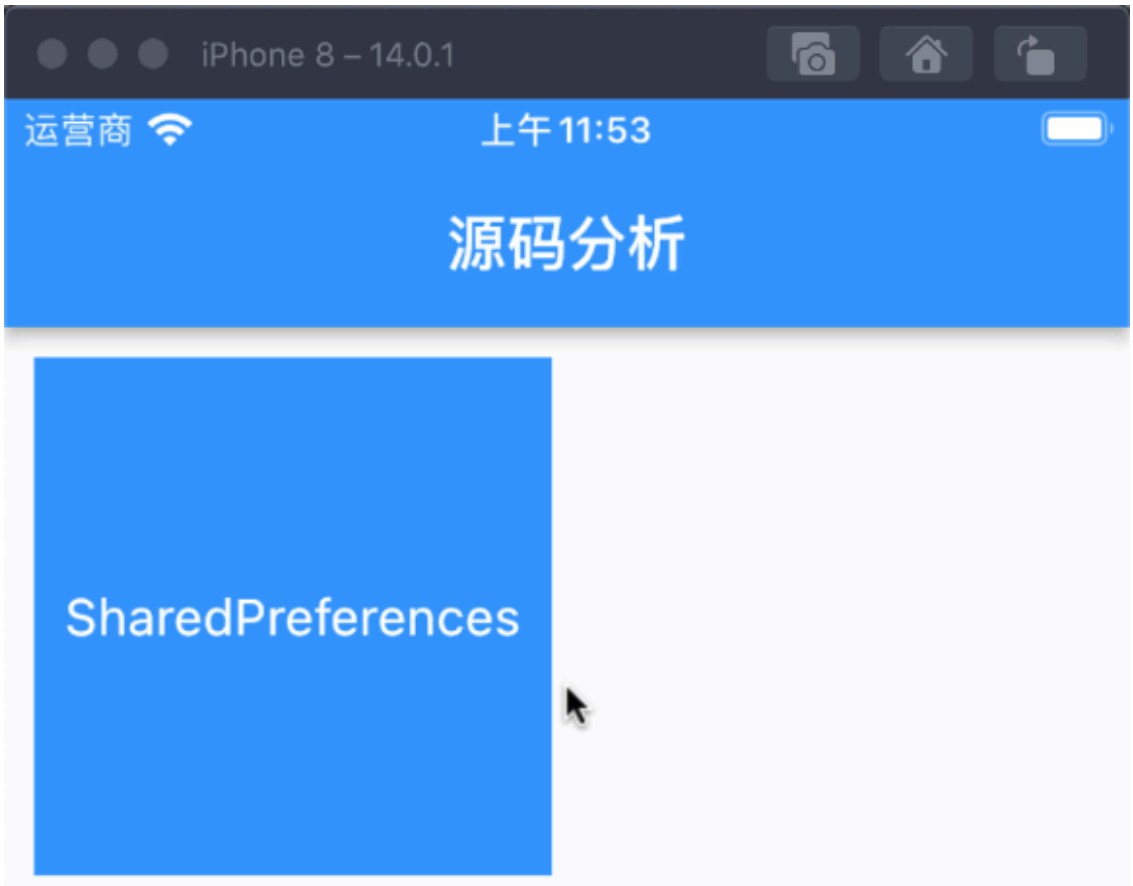
```
import 'package:flutter/material.dart';
import 'package:shared_preferences/
shared_preferences.dart';

class SharedPreferencesExample extends StatefulWidget {
  @override
  _SharedPreferencesExampleState createState() =>
_SharedPreferencesExampleState();
}

class _SharedPreferencesExampleState extends
State<SharedPreferencesExample> {


  SharedPreferences? sharedPreferences;

  // 设置持久化数据
```

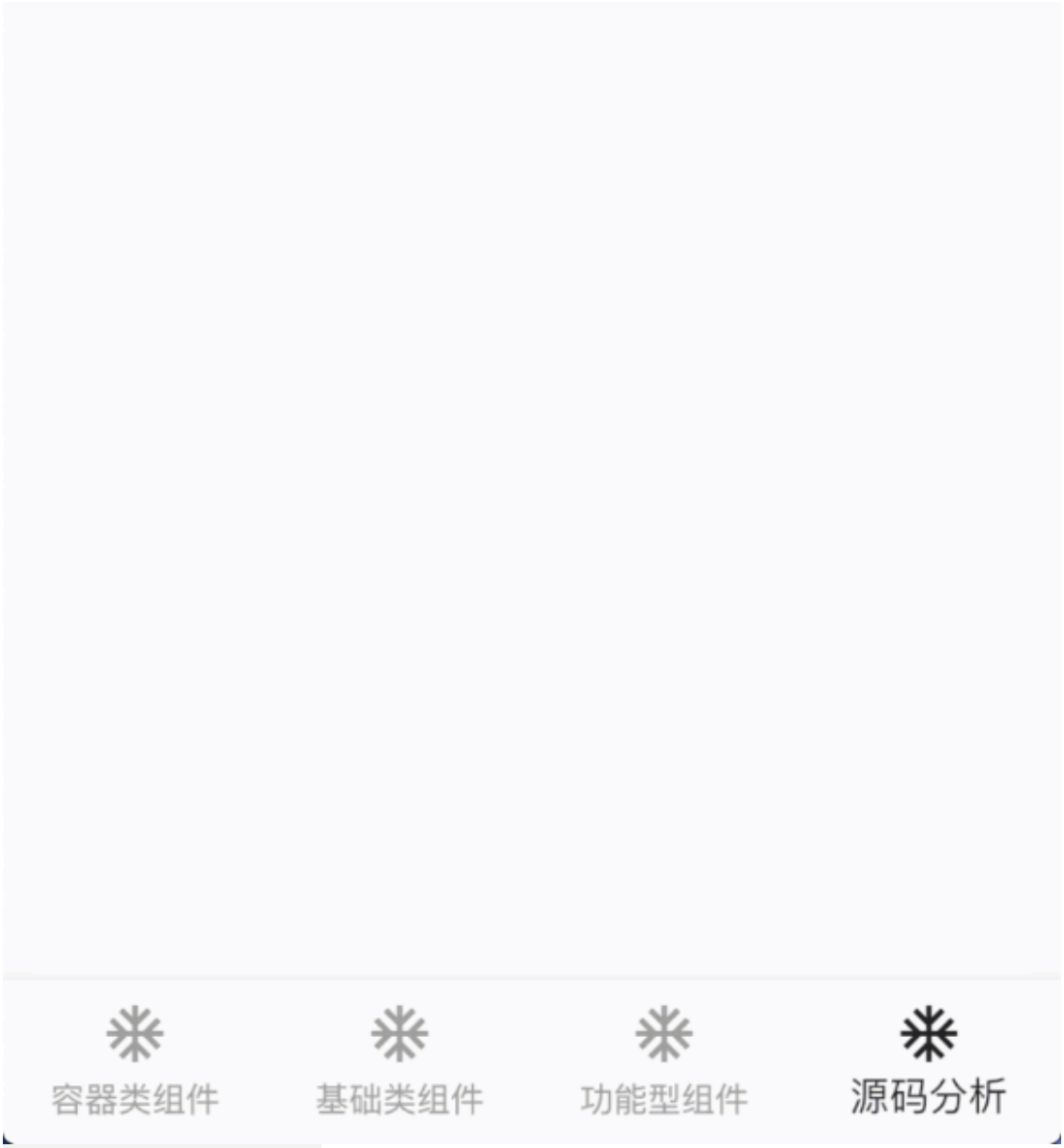
```
  void _setData() async {
    // 实例化
    sharedPreferences = await
SharedPreferences.getInstance();

    // 设置string类型
    await sharedPreferences?.setString("name", "Jimi");

    // 设置int类型
    await sharedPreferences?.setInt("age", 18);

    // 设置bool类型
    await sharedPreferences?.setBool("isTeacher", true);

    // 设置double类型
    await sharedPreferences?.setDouble("height", 1.88);

    // 设置string类型的数组
    await sharedPreferences?.setStringList("action", ["吃
饭", "睡觉", "打豆豆"]);

    setState(() {});
  }


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("SharedPreferences"),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _setData,
        child: Icon(Icons.add),
      ),
```

```
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            Text("名字: $
{sharedPreferences?.getString("name") ?? ""}",
                style: TextStyle(
                  color: Colors.blue,
                  fontSize: 20
                ),
              ),
            SizedBox(height: 20,),
            Text("年龄: $
{sharedPreferences?.getInt("age") ?? ""}",
                style: TextStyle(
                  color: Colors.red,
                  fontSize: 20
                ),
              ),
            SizedBox(height: 20,),
            Text("是老师吗?: $
{sharedPreferences?.getBool("isTeacher") ?? ""}",
                style: TextStyle(
                  color: Colors.orange,
                  fontSize: 20
                ),
              ),
            SizedBox(height: 20,),
            Text("身高: $
{sharedPreferences?.getDouble("height") ?? ""}",
                style: TextStyle(
                  color: Colors.pink,
                  fontSize: 20
                ),
```

```
                ),
            SizedBox(height: 20,),
            Text("我正在: $
{sharedPreferences?.getStringList("action") ?? ""}",
                style: TextStyle(
                  color: Colors.purple,
                  fontSize: 20
                ),
              ),
          ],
        ),
      ),
    );
  }
}
```

效果如下





`shared_preferences`辅助操作

获取持久化数据中所有存入的key

```
List<String> keys =
sharedPreferences?.getKeys().toList() ?? [];
print(keys);
```

// 控制台输出

```
[name, age, isTeacher, height, action]
```

判断持久化数据中是否包含某个 key

```
bool isContainKey =
sharedPreferences?.containsKey("name") ?? false;
print(isContainKey);
```

// 控制台输出

```
flutter: true
```

删除持久化数据中某个 key

```
bool isRemoveKey = await
sharedPreferences?.remove("name") ?? false;
print(isRemoveKey);
```

// 控制台输出

```
flutter: true
```

清除所有持久化数据

```
bool isClearAllKey = await sharedPreferences?.clear() ??
false;
print(isClearAllKey);
```

// 控制台输出

```
flutter: true
```

重新加载所有数据（仅重载运行时）

```
await sharedPreferences?.reload();
```

本篇主要讲`shared_preferences`的使用，下篇就来讲讲

`shared_preferences`的源码和封装