



SharedPreferences 使用总结

适用范围：保存少量的数据，且这些数据的格式非常简单：字符串型、基本类型的值。比如应用程序的各种配置信息（如是否打开音效、是否使用震动效果、小游戏的玩家积分等），解锁口令密码等。

核心原理：保存基于 XML 文件存储的 key-value 键值对数据，通常用来存储一些简单的配置信息。通过 DDMS 的 File Explorer 面板，展开文件浏览树，很明显 SharedPreferences 数据总是存储在 /data/data/<package name>/shared_prefs 目录下。SharedPreferences 对象本身只能获取数据而不支持存储和修改，存储修改是通过 SharedPreferences.edit() 获取的内部接口 Editor 对象实现。SharedPreferences 本身是一个接口，程序无法直接创建 SharedPreferences 实例，只能通过 Context 提供的 getSharedPreferences(String name, int mode) 方法来获取 SharedPreferences 实例，该方法中 name 表示要操作的 xml 文件名，第二个参数具体如下：

Context.MODE_PRIVATE：指定该 SharedPreferences 数据只能被本应用程序读、写。

Context.MODE_WORLD_READABLE：指定该 SharedPreferences 数据能被其他应用程序读，但不能写。

Context.MODE_WORLD_WRITEABLE：指定该 SharedPreferences 数据能被其他应用程序读、写 Editor 有如下主要重要方法：

SharedPreferences.Editor clear()：清空 SharedPreferences 里所有数据

SharedPreferences.Editor putXxx(String key, xxx value)：向 SharedPreferences 存入指定 key 对应的数据，其中 xxx 可以是 boolean, float, int 等各种基本类型据

SharedPreferences.Editor remove()：删除 SharedPreferences 中指定 key 对应的数据项

boolean commit()：当 Editor 编辑完成后，使用该方法提交修改



实际案例：运行界面如下



这里只提供了两个按钮和一个输入文本框，布局简单，故在此不给出界面布局文件了，程序核心代码如下：

```
class View0c1 implements View.OnClickListener {

    @Override
    public void onClick(View v) {

        switch(v.getId()) {
            case R.id.btnSet:
                //步骤 1：获取输入值
                String code = txtCode.getText().toString().trim();
                //步骤 2-1：创建一个 SharedPreferences.Editor 接口对象，lock 表示要写入的 XML 文件名，MODE_WORLD_WRITEABLE 写操作
                SharedPreferences.Editor editor =
                getSharedPreferences("lock", MODE_WORLD_WRITEABLE).edit();
                //步骤 2-2：将获取过来的值放入文件
                editor.putString("code", code);
                //步骤 3：提交
                editor.commit();
                Toast.makeText(getApplicationContext(), "口令设置成功", Toast.LENGTH_LONG).show();
            }
        }
    }
}
```



```
        break;
    case R.id.btnGet:
        //步骤 1: 创建一个 SharedPreferences 接口对象
        SharedPreferences read = getSharedPreferences("lock",
MODE_WORLD_READABLE);
        //步骤 2: 获取文件中的值
        String value = read.getString("code", "");
        Toast.makeText(getApplicationContext(), "口令为:
"+value, Toast.LENGTH_LONG).show();
        break;
    }
}
```

读写其他应用的 SharedPreferences: 步骤如下

1、在创建 SharedPreferences 时, 指定 MODE_WORLD_READABLE 模式, 表明该 SharedPreferences 数据可以被其他程序读取。

2、创建其他应用程序对应的 Context:

Context pvCount = createPackageContext("com. tony. app",
Context. CONTEXT_IGNORE_SECURITY);这里的 com. tony. app 就是其他程序的包名。

3、使用其他程序的 Context 获取对应的 SharedPreferences

SharedPreferences read = pvCount.getSharedPreferences("lock",
Context. MODE_WORLD_READABLE);

4、如果是写入数据, 使用 Editor 接口即可, 所有其他操作均和前面一致。

SharedPreferences 对象与 SQLite 数据库相比, 免去了创建数据库, 创建表, 写 SQL 语句等诸多操作, 相对而言更加方便, 简洁。但是



SharedPreferences 也有其自身缺陷，比如其只能存储 `boolean`, `int`, `float`, `long` 和 `String` 五种简单的数据类型，比如其无法进行条件查询等。所以不论 SharedPreferences 的数据存储操作是如何简单，它也只能是存储方式的一种补充，而无法完全替代如 `SQLite` 数据库这样的其他数据存储方式。