使用公有外部存储

1.1 知识解析

由于外部存储可能不可用，因此，在访问它之前，应始终确认其状态。 可以通过调用 Environment.getExternalStorageState() 查询外部存储状态。 如果返回的状态为 Environment.MEDIA_MOUNTED，那么可以对文件进行读写。 可能的状态包括：

- MEDIA_MOUNTED
- MEDIA_UNMOUNTED
- MEDIA_MOUNTED_READ_ONLY
- MEDIA_REMOVED

外部存储文件分 2 种：

公共文件

可供其他应用和用户自由使用的文件。 当用户卸载应用时，用户应仍可以使用这些文件。

例如，应用拍摄的照片或音乐程序下载的 mp3 文件等。

私有文件

本属于某个应用且应在用户卸载此应用时删除的文件。尽管这些文件在技术上可被用户和其他应用访问（因为它们在外部存储上），它们是实际上不向这个应用之外的用户提供的文件。当用户卸载应用时，系统会删除应用外部专用目录中的所有文件。

例如，应用下载的其他资源或临时介质文件。

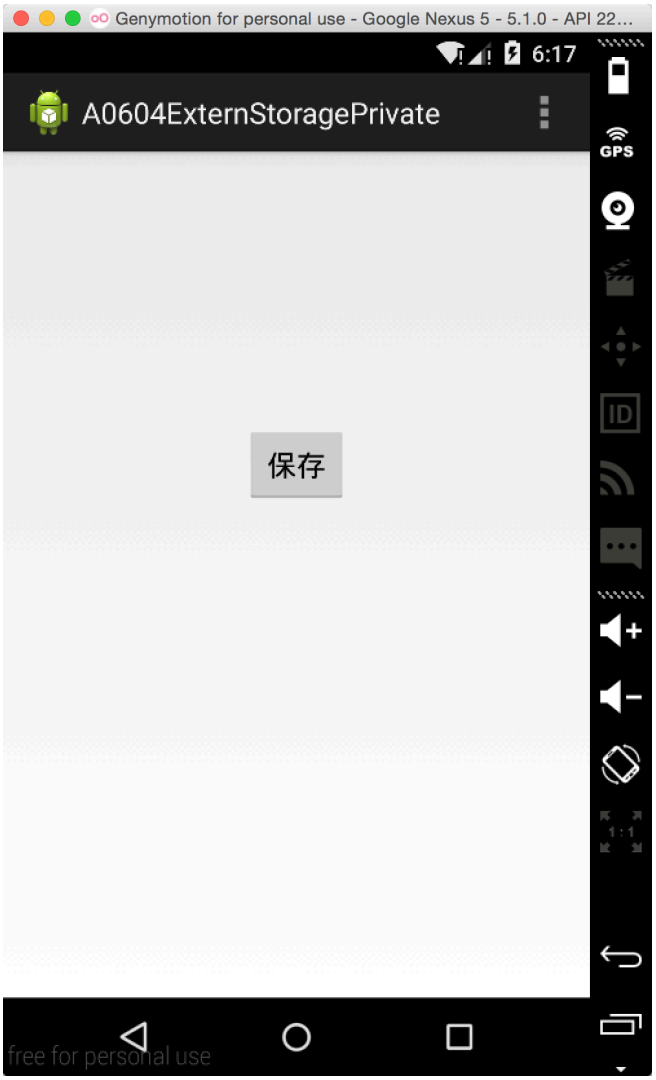
如果要使用外部存储上的公共文件，使用 Environment.getExternalStoragePublicDirectory(String type) 方法获取表示外部存储上相应目录的 File 。该方法使用指定想要保存以便它们可以与其他公共文件在逻辑上组织在一起的文件类型的参数，比如 Environment.DIRECTORY_MUSIC 或 DIRECTORY_PICTURES。

注意使用公共外部存储文件需要读写权限：

android.permission.WRITE_EXTERNAL_STORAGE、

android.permission.READ_EXTERNAL_STORAGE(API 16 新增)

1.2 功能演示



1.3 实战操作

```
File docFile = new File(file, "mydoc.txt");
```

```
            try {
                FileOutputStream fos = new FileOutputStream(docFile, true);
                String str = "This is a document";
                byte[] buffer = str.getBytes();
                fos.write(buffer);
                fos.close();
            } catch (FileNotFoundException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
```

1.4 职业素质

其实安卓文件的操作和 java 在 pc 环境下的操作并无二致，之所以需要单独讲解是因为安卓系统提供了不同于 pc 的访问文件系统根路径的 api，同时对一个应用的私有文件做了统一的管理。根据我的经验，初学者在这部分感到很容易混淆内部存储和外部存储两个概念。