# TODO

<https://blog.csdn.net/u011146511/article/details/80486255>

<https://blog.csdn.net/u011814346/article/details/80196622>

StorageManager.getVolumeList(排序规则是啥？

内部优先？？

# Environment

获取环境变量

*/\*\*  
 \* Return root of the "system" partition holding the core Android OS.  
 \* Always present and mounted read-only.  
 \*/***public static** File getRootDirectory() {  
 **return** DIR\_ANDROID\_ROOT;  
}

**private static final** File DIR\_ANDROID\_ROOT = getDirectory(ENV\_ANDROID\_ROOT, **"/system"**);

**private static final** String ENV\_ANDROID\_ROOT = **"ANDROID\_ROOT"**;

**static** File getDirectory(String variableName, String defaultPath) {  
 String path = System.getenv(variableName);  
 **return** path == **null** ? **new** File(defaultPath) : **new** File(path);  
}

## 系统环境变量

用常用的linux命令就好了

bullhead:/ $ echo ${ANDROID\_ROOT}

/system

## throwIfUserRequired

在系统进程中，应该指定一个user\_id来调用具体接口

### 保存日志

Zs

06-02 20:12:42.357 4355-4355/system\_process E/Environment: Path requests must specify a user by using UserEnvironment

java.lang.Throwable

at android.os.Environment.throwIfUserRequired(Environment.java:913)

at android.os.Environment.getExternalStorageDirectory(Environment.java:449)

### 调用位置

#### setUserRequired

SystemServer.run()

*// Within the system server, it is an error to access Environment paths without  
// explicitly specifying a user.*Environment.setUserRequired(**true**);

*/\*\* {****@hide****} \*/***public static void** setUserRequired(**boolean** userRequired) {  
 sUserRequired = userRequired;  
}

#### getExternalStorageDirectory

**public static** File getExternalStorageDirectory() {  
 throwIfUserRequired();  
 **return** sCurrentUser.getExternalDirs()[0];  
}

#### throwIfUserRequired

**private static void** throwIfUserRequired() {  
 **if** (sUserRequired) {  
 Log.wtf(TAG, **"Path requests must specify a user by using UserEnvironment"**,  
 **new** Throwable());  
 }  
}

## UserEnvironment

**public static** File getExternalStorageDirectory() {  
 *throwIfUserRequired*();  
 **return** *sCurrentUser*.getExternalDirs()[0];  
}

因此排在第一位的就存第一位的

老的数据，怎么办？？？数据升级有问题么？

**protected final boolean externalSdFirst** = **true**;  
**public** File[] getExternalDirs() {  
 **final** StorageVolume[] volumes = StorageManager.getVolumeList(**mUserId**,  
 StorageManager.FLAG\_FOR\_WRITE);  
 **final** File[] files = **new** File[volumes.**length**];  
 **for** (**int** i = 0; i < volumes.**length**; i++) {  
 **if** (**externalSdFirst** && files[i].getPath().contains(**"external\_sd"**)){  
 **for** (**int** j = 0; j <= i; j++){  
 files[j+1] = files[j];  
 }  
 files[0] = volumes[i].getPathFile();  
 }**else** {  
 files[i] = volumes[i].getPathFile();  
 }  
 }  
 **return** files;  
}

## 定制

支持用户选择sd卡当做默认数据存储

可以使用db方式么？为啥？为何用属性系统