

愿做您成功路上的基石!

——百杰科技

Android 统一接口

说明文档

V1.3

2025-8



www.szbaijie.com

www.szbaijie.cn

修订记录

日期	版本	修订说明	作者
2022.07.28	V1.0	新建	范孝文
2023.06.08	V1.1	增加 wifi 列表信息等	范孝文
2023.11.30	V1.2	增加设置 wifi, 设置网络优先级	刘丰
2025.8.12	V1.3	新增加设置 root, 204 服务器, 旋转角度, 使能悬浮窗	刘丰

目录

1 如何使用.....	1
2 系统.....	2
2.1 获取设备型号.....	2
2.2 获取 JAR 版本.....	2
2.3 获取设备唯一 ID.....	2
2.4 获取系统版本.....	3
2.5 获取系统 SDK 版本.....	3
2.6 获取内部存储大小.....	3
2.7 获取剩余运行内存.....	4
2.8 获取内部 SD 卡路径.....	4
2.9 获取外部 SD 卡路径.....	5
2.10 获取 U 盘路径.....	5
2.11 关机.....	5
2.12 重启.....	6
2.13 判断包名对应的 APP 是否存在.....	6
2.14 打开系统设置界面.....	6
2.15 打开系统 Wifi 设置界面.....	7
2.16 设置 GPIO 输出值.....	7
2.17 获取 GPIO 值.....	7
2.18 设置 GPIO 方向.....	8
2.19 获取 GPIO 的方向值.....	8

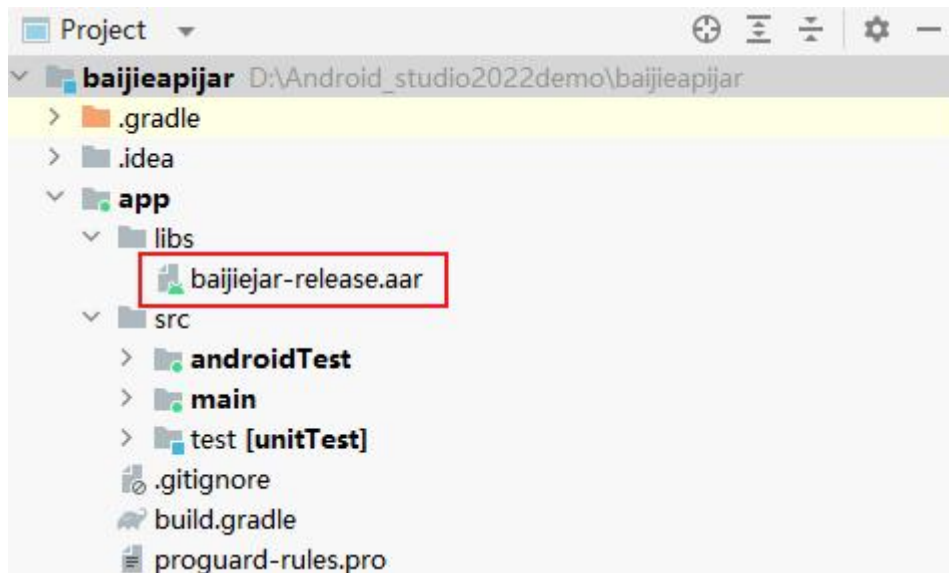
2.20 设置开机自启动.....	8
2.21 获取设置的开机自启动 APP 包名和 Activity 类名.....	9
2.22 启动另一个 APP.....	9
2.23 判断文件是否存在.....	10
2.24 静默安装 APK.....	10
2.25 卸载 APP.....	10
2.26 静默安装 APK 并且重启系统.....	11
2.27 安装 APP 并启动安装的 APP.....	11
2.28 守护进程, 保持置顶状态.....	11
2.29 取消守护进程, 取消置顶状态.....	12
2.30 获取系统最大音量.....	12
2.31 获取系统当前音量.....	12
2.32 增大音量.....	13
2.33 降低音量.....	13
2.34 设置音量.....	13
2.35 设置系统桌面.....	14
2.36 删除系统桌面.....	14
2.37 设置 usb0 host 模式.....	14
2.38 设置 usb0 device 模式.....	15
2.39 设置桌面旋转.....	15
2.40 设置开机 logo.....	16
2.41 设置 204 服务器.....	16

2.42 设置使能悬浮窗.....	17
3 网络.....	18
3.1 获取指定网络的 IPV4 地址.....	18
3.2 判断 wifi 是否打开.....	18
3.3 获取当前连接 wifi 名称.....	18
3.4 获取 wifi 列表信息.....	19
3.5 设置当前连接 WIFI.....	19
3.6 设置网络优先级.....	20
4 时间.....	21
4.1 获取系统日期.....	21
4.2 获取系统时间.....	21
4.3 设置系统时间.....	21
4.4 判断系统的时间是否自动获取.....	22
5 显示.....	23
5.1 获取 Android 系统的屏幕分辨率.....	23
5.2 获取 Android 设备显示分辨率 Y 轴像素.....	23
5.3 获取 Android 设备显示分辨率 X 轴像素.....	23
5.4 获取屏幕显示密度.....	24
5.5 设置屏幕显示密度.....	24
5.6 获取最大亮度值.....	24
5.7 增大亮度值.....	25
5.8 降低亮度值.....	25

5.9 获取系统当前亮度值.....	25
5.10 设置系统亮度值.....	26
6 关于 root.....	27
6.1 整机 root.....	27

1 如何使用

将 jar 包或者 aar 包放进需要使用的模块资源目录下，如图



在 build.gradle 或者 build.gradle.kt（新版本 android studio）中加入依赖

```
implementation fileTree(dir: 'libs', include: ['*.aar', '*.jar'],  
exclude: [])
```

这样就可以导入我们做好的类了

```
BaijieManager bm = new BaijieManager();  
bm.getDeviceVersion();//获取设备型号
```

2 系统

2.1 获取设备型号

函数：public static String getDeviceVersion();

描述：获取 Android 设备型号。

用途：可以通过读取这个 api 适配不同的板子，写不同的功能代码。

参数说明：

参数/返回值	类型	说明	数据
返回值	String	Android 设备的设备型号	QUAD-CORE A133 c4

示例：

```
String deviceVersion = baijieManager.getDeviceVersion();  
//deviceVersion 为获取到的设备型号
```

2.2 获取 JAR 版本

函数：public String getJARVersion();

描述：获取 jar 版本号。

用途：当用户拿到的 jar 版本和厂家公布的 jar 不一致时，建议更新

参数说明：

参数/返回值	类型	说明	数据
返回值	String	jar 版本号	如 1.0

示例：

```
String JARVersion = baijieManager.getJARVersion();  
//JARVersion 为获取到的设备型号
```

2.3 获取设备唯一 ID

函数：public String getDeviceID();

描述：获取 Android 设备唯一序列号

参数说明：

参数/返回值	类型	说明	数据
返回值	String	序号	

示例:

```
String deviceID = baijieManager.getDeviceID();  
//deviceID 为获取到的设备序号
```

2.4 获取系统版本

函数: `public static String getAndroidVersion();`

描述: 获取 Android 设备系统版本号

用途: 可以通过读取这个 api 适配不同的操作系统版本, 写不同的功能代码。

参数说明:

参数/返回值	类型	说明	数据
返回值	String	Android 设备系统版本号	10

示例:

```
String AndroidVersion = baijieManager.getAndroidVersion();  
//AndroidVersion 为获取到的设备系统版本号
```

2.5 获取系统 SDK 版本

函数: `public int getSDKVersion();`

描述: 获取 Android 设备系统 SDK 版本号

参数说明:

参数/返回值	类型	说明	数据
返回值	int	Android 设备系统 SDK 版本号	29

示例:

```
int SDKVersion = baijieManager.getSDKVersion();  
//SDKVersion 为获取到的设备系统 SDK 版本号
```

2.6 获取内部存储大小

函数: `public long getTotalInternalMemorySize();`

描述: 获取内部存储空间总大小, 相当于电脑的硬盘空间大小, 单位: 字节

用途：可以写一些监听内存的代码，转换 GB 需要将返回的结果/1024/1024/1024

参数说明：

参数/返回值	类型	说明	数据
返回值	long	Android 设备内部存储大小（单位：字节）	

示例：

```
long totalInternalMemorySize =  
baijieManager.getTotalInternalMemorySize();  
//totalInternalMemorySize 为获取到的内部存储大小（字节）
```

2.7 获取剩余运行内存

函数：public long getFreeMemorySize();

描述：获取剩余的运行内存，单位：字节。

用途：可用于监听 app 内存使用情况，用来查看是否有内存泄漏。

参数说明：

参数/返回值	类型	说明	数据
返回值	long	设备剩余运行内存（单位：字节）	490176512

示例：

```
long freeMemorySize=baijieManager.getFreeMemorySize();  
//freeMemorySize 为获取到的剩余运行内存（字节）
```

2.8 获取内部 SD 卡路径

函数：public String getInternalSDCardPath();

描述：获取 Android 设备内部 sdcard 路径

参数说明：

参数/返回值	类型	说明	数据
返回值	String	内部 SD 卡路径	/sdcard

示例：

```
String internalSDCardPath = baijieManager.getInternalSDCardPath();  
//internalSDCardPath 为获取到的内部 SD 卡路径
```

2.9 获取外部 SD 卡路徑

函数: `public String getExternalSDCardPath();`

描述: 获取 Android 设备外部 sdcard 路径, 比如说 TF 卡。

参数说明:

参数/返回值	类型	说明	数据
返回值	String	外部 SD 卡路徑	/storage/0000-0000

示例:

```
String externalSDCardPath = baijieManager.getExternalSDCardPath();  
//externalSDCardPath 为获取到的外部 SD 卡路徑
```

2.10 获取 U 盘路径

函数: `public String getUsbStoragePath();`

描述: 获取 Android 设备外部 sdcard 路径, 比如说 TF 卡。

备注: 因为监听系统广播的 `media_mounted` 返回的路径各个平台都不统一, 所以如果想获取精确的 U 盘路径, 推荐使用此接口。当 U 盘插入一段时间后, 才能够返回正确的值。不插入 U 盘则返回 `null`。

参数说明:

参数/返回值	类型	说明	数据
返回值	String	获取 U 盘路径	/storage/0000-0000

示例:

```
String usbStoragePath = baijieManager.getUsbStoragePath();  
//usbStoragePath 为获取到的获取 U 盘路径
```

2.11 关机

函数: `public void shutdown();`

描述: 使 Android 系统进入关机状态。

示例:

```
baijieManager.shutdown();  
//Android 系统进入关机状态
```

2.12 重启

函数：public void reboot(int delay);

描述：重启 Android 系统。

说明：该接口在选在关机时会保存用户数据，以防用户数据丢失

参数说明：

参数/返回值	类型	说明	数据
delay	int	延迟时间，单位：秒，如不需要延时，填入 0 即可	

示例：

```
baijieManager.reboot(1000);  
//1 秒后进入重启
```

2.13 判断包名对应的 APP 是否存在

函数：public boolean isAppExist(String pkgName);

描述：判断包名对应的 APP 是否存在设备中

参数说明：

参数/返回值	类型	说明	数据
pkgName	String	需要判断的 app 包名	"com.android.settings"
返回值	boolean	True:存在, false:不存在	true

示例：

```
boolean result = baijieManager.isAppExist("com.android.settings");  
//result 为判断 APP 是否存在的返回值
```

2.14 打开系统设置界面

函数：public void startSettings();

描述：打开系统设置界面

示例：

```
baijieManager.startSettings();
```

2.15 打开系统 Wifi 设置界面

函数：public void startWifiSettings();

描述：打开系统 Wifi 设置界面。

示例：

```
baijieManager.startWifiSettings();
```

2.16 设置 GPIO 输出值

函数：public void setGpioValue(String port, int value);

描述：将对应的 GPIO 设置为输出模式，并设置输出电平。

参数说明：

参数/返回值	类型	说明	数据
port	String	GPIO 口，具体 IO 口号参考对应列表	例：“B8”
value	int	1:输出高电平 0:输出低电平	0 或 1

示例：

```
baijieManager.setGpioValue(“B8”,1);  
//设置 GPIO_B8 输出高电平
```

2.17 获取 GPIO 值

函数：public int getGpioValue(String port, String direction);

描述：获取指定 GPIO 的值。

参数说明：

参数/返回值	类型	说明	数据
port	String	GPIO 口，具体 IO 口号参考对应列表	例:”B8”
direction	String	in:输入 out:输出	例:”out”
返回值	int	GPIO 的输出值	

示例：

```
int Value = baijieManager.getGpioValue(“B8”,”out”);  
//Value 为获取到的 GPIO_B8 输出值
```

2.18 设置 GPIO 方向

函数: `public void setGpioDirection(String port, String direction);`
描述: 设置 GPIO 的方向

参数说明:

参数/返回值	类型	说明	数据
port	String	GPIO 口, 具体 IO 口号参考对应列表	例: "B8"
direction	String	in:输入 out:输出	例: "out"

示例:

```
baijieManager.setGpioDirection("B8", "out");  
//设置 GPIO_B8 为输出状态
```

2.19 获取 GPIO 的方向值

函数: `public String getGpioDirection(String port);`
描述: 获取 GPIO 的方向值

参数说明:

参数/返回值	类型	说明	数据
port	String	GPIO 口, 具体 IO 口号参考对应列表	例: "B8"
返回值	String	in:输入 out:输出	例: "out"

示例:

```
String value = baijieManager.getGpioDirection("B8");  
//value 为获取到 GPIO_B8 的方向值
```

2.20 设置开机自启动

函数: `public void setBootPackageActivity(String pkgName, String pkgActivity);`

描述: 设置 Android 系统开机自启动 APP 包名和 Activity, 参数为空时取消开机自启动。

参数说明:

参数/返回值	类型	说明	数据
pkgName	String	APP 包名, 取消参数 填空即可	"com.android.settings"
pkgActivity	String	APP 的 Activity 类名,	".SubSettings"

		取消参数填空即可	
--	--	----------	--

示例:

```
baijieManager.setBootPackageActivity(“com.android.settings”, “.SubSettings”);  
//设置【系统设置】为开机自启动
```

2.21 获取设置的开机自启动 APP 包名和 Activity 类名

函数: `public String getBootPackageName();`

描述: 获取 Android 设备开机自启动的 APP 包名和类名

```
示例: String packageName = “”;  
String pkgActivity = “”;  
String pkgNameActivity =  
baijieManager.getBootPackageName();  
String[] parts = pkgNameActivity.split("/\\.");  
packageName = parts[0];  
pkgActivity = parts[1];  
//packageName 为包名, pkgActivity 为类名
```

2.22 启动另一个 APP

函数: `public void startActivity(String pkgName, String pkgActivity);`

描述: 启动另一个 APP

备注: 标准做法的一种封装而已, 更易于使用。

参数说明:

参数/返回值	类型	说明	数据
pkgName	String	APP 包名	com.android.settings
pkgActivity	String	APP 的 Activity 类名	com.android.settings.Settings

示例:

```
baijieManager.startActivity(“com.android.settings”, “com.android.settings.Settings”);  
//打开【系统设置】界面
```

2.23 判断文件是否存在

函数：public boolean isExist(String path);

描述：判断指定路径名称文件是否存在

参数说明：

参数/返回值	类型	说明	数据
path	String	需要判断的文件路径	/sdcard/test.apk
返回值	boolean	True:存在, false:不存在	true

示例：

```
boolean result = baijieManager.isExist(j);  
//result 为判断文件是否存在的结果
```

2.24 静默安装 APK

函数：public void installAppSilent(String filePath)

描述：静默安装指定路径的 APK 文件

说明：仅限于有 root 权限的固件使用。

参数说明：

参数/返回值	类型	说明	数据
filePath	String	APK 文件路径	/sdcard/test.apk

示例：

```
baijieManager.installAppSilent(“/sdcard/test.apk”);  
//静默安装 test.apk
```

2.25 卸载 APP

函数：public void uninstallAppSilent(String packageName);

描述：卸载指定包名的 APP。

参数说明：

参数/返回值	类型	说明	数据
packageName	String	指定 APP 包名	“com.dc.test”

示例：

```
baijieManager.uninstallAppSilent(("com.dc.test");  
//卸载 test.apk
```

2.26 静默安装 APK 并且重启系统

函数: `public void installAppSilentAndRebootSystem(String filePath, int reboot_in_sec);`

描述: 静默安装 APK 并且重启系统

参数说明:

参数/返回值	类型	说明	数据
filePath	String	APK 文件路径	/sdcard/test.apk
reboot_in_sec	int	安装完成后延迟多少秒再重启	1

示例:

```
baijieManager.installAppSilentAndRebootSystem("/sdcard/test.apk", 1);  
//安装/sdcard/目录下的 test.apk, 并且在 1 秒后重启 Android 系统
```

2.27 安装 APP 并启动安装的 APP

函数: `public void installAppAndStartUp(String filePath, String pkgName);`

描述: 安装 APK, 并且在安装完成后启动所安装的 APP

参数说明:

参数/返回值	类型	说明	数据
filePath	String	APK 文件路径	/sdcard/test.apk
pkgName	String	要启动的 APP 包名	com.dc.test

示例:

```
baijieManager.installAppAndStartUp("/sdcard/test.apk", "com.dc.test")  
;  
//安装/sdcard/目录下的 test.apk, 并且安装完成后启动 test
```

2.28 守护进程, 保持置顶状态

函数: `public void keepActivity(String pkgName, String ActivityName);`

描述: 在程序被切换到后台或关闭时, 自动返回或启动该 app, 保持该 app 在前端。当需要打开系统设置界面或运行其他 app 暂时进行其他设置时, 请先调用取消接口或者把 app 后台撤了, 否则会一直保持 app 置顶导致无法操作。

参数说明:

参数/返回值	类型	说明	数据
pkgName	String	需要保持置顶的 app 包名	"com.dc.test"
ActivityName	String	需要保持置顶的 app 活动名	MainActivity

示例:

```
baijieManager.keepActivity(" com.dc.test", " MainActivity" );  
//保持包名为"com. dc. test"的 APP 置顶状态
```

2.29 取消守护进程, 取消置顶状态

函数: public void unKeepActivity();

描述: 取消守护进程, 取消置顶状态

示例:

```
baijieManager.unKeepActivity();  
//取消置顶状态
```

2.30 获取系统最大音量

函数: public int getSystemMaxVolume();

描述: 获取 Android 系统最大媒体音量值

参数说明:

参数/返回值	类型	说明	数据
返回值	int	Android 系统最大媒体音量值	15

示例:

```
int systemMaxVolume = baijieManager.getSystemMaxVolume();  
//systemMaxVolume 为获取到的 Android 系统最大媒体音量值
```

2.31 获取系统当前音量

函数: public int getSystemCurrenVolume();

描述: 获取 Android 系统当前媒体音量值

参数说明:

参数/返回值	类型	说明	数据
返回值	int	Android 系统当前媒体音量值	0-15

示例:

```
int systemCurrenVolume = baijieManager.getSystemCurrenVolume();  
//systemCurrenVolume 为获取到的 Android 系统当前媒体音量值。
```

2.32 增大音量

函数: public int setRaiseSystemVolume();

描述: 设置 Android 系统媒体音量值+1

参数说明:

参数/返回值	类型	说明	数据
返回值	int	0: 设置成功 -1: 设置失败	0

示例:

```
int flag = baijieManager.setRaiseSystemVolume();  
//设置 Android 系统媒体音量值+1。
```

2.33 降低音量

函数: public int setLowerSystemVolume();

描述: 设置 Android 系统媒体音量值-1

参数说明:

参数/返回值	类型	说明	数据
返回值	int	0: 设置成功 -1: 设置失败	0

示例:

```
int flag = baijieManager.setLowerSystemVolume();  
//设置 Android 系统媒体音量值-1。
```

2.34 设置音量

函数: public int setSystemVolumeIndex(int index);

描述: 设置 Android 系统媒体音量值。

参数说明:

参数/返回值	类型	说明	数据
index	int	设置 Android 系统媒体音量值	0-15
返回值	int	0: 设置成功 -1: 设置失败	0

示例:

```
returnValue = baijieManager.setSystemVolumeIndex(10);  
//设置 Android 系统媒体音量值为 10。
```

2.35 设置系统桌面

函数: public void setDefaultLauncher(String defPackageName);

描述: 设置 Android 系统桌面。需要先删除原有的默认系统桌面 com.android.launcher3

参数说明:

参数/返回值	类型	说明	数据
pkgName	String	APP 包名, 取消参数 填空即可	"com.watmind.xn"

示例:

```
baijieManager.setBootPackageActivity( "com.watmind.xn" );  
//设置系统桌面
```

2.36 删除系统桌面

函数: public void delDefaultLauncher(String defPackageName);

描述: 删除 Android 系统桌面。

参数说明:

参数/返回值	类型	说明	数据
pkgName	String	Launcher 包名	"com.watmind.xn"

示例:

```
baijieManager.delDefaultLauncher( "com.watmind.xn" );  
//设置系统桌面
```

2.37 设置 usb0 host 模式

函数: public boolean setUsbUse2();

描述：设置 usb0 host 模式。

参数说明：

参数/返回值	类型	说明
是否成功	Boolean	设置 usb0 host

示例：

```
baijieManager.setUsbUse2();
```

注意：a523 ,t527 usb0 host 后才能使用 usb down 硬件接口

2.38 设置 usb0 device 模式

函数：public boolean setUsbDel2();

描述：设置 usb0 device 模式。

参数说明：

参数/返回值	类型	说明
是否成功	Boolean	设置 usb0 device

示例：

```
baijieManager.setUsbDel2();
```

2.39 设置桌面旋转

函数：public void setRotation(int number);

描述：设置 Android 系统桌面旋转。

参数说明：

参数/返回值	类型	说明	数据
旋转角度	Int	参数为 1,2,3,4。1 为 0 度, 2 为 90 度,3 为 180 度, 4 为 270 度	2

示例：

```
baijieManager.setRotation(1);
```

```
//设置桌面旋转，重启生效
```

2.40 设置开机 logo

```
public boolean setBootlogo(String path);
```

描述：设置 Android 开机 logo。

参数说明：

参数/返回值	类型	说明	数据
path	String	设置图片路径, 图片格式为 bmp	"/storage/emulated/0/bootlogo.bmp"

示例：

```
baijieManager.setBootlogo(“/storage/emulated/0/bootlogo.bmp”);  
//设置开机 logo, 重启生效
```

2.41 设置 204 服务器

```
public boolean setDetectingServers(String[] servers);
```

描述：设置 Android 检查网络的 204 服务器。

参数说明：

参数/返回值	类型	说明	数据
Servers	String[]	设置 Service	{"http://connectivitycheck.platform.hicloud.com/generate_204","https://connectivitycheck.platform.hicloud.com/generate_204"}

示例:

```
String[] servres =  
{ "http://connectivitycheck.platform.hicloud.com/generate_204", "  
https://connectivitycheck.platform.hicloud.com/generate_204" };  
baijieManager.setDetectingServers(servres);  
//设置网络检查服务器，重启网络生效
```

2.42 设置使能悬浮窗

```
public boolean setVideoPane(boolean isVideoPane);
```

描述: 设置 Android 是否使能悬浮窗。

参数说明:

参数/返回值	类型	说明	数据
isVideoPane	Boolean	设置是否使能	true

示例:

```
baijieManager.setVideoPane(true);  
//设置使能悬浮窗，重启生效
```

3 网络

3.1 获取指定网络的 IPV4 地址

函数：public String getIPv4(String nettype);

描述：获取 Android 设备指定网络的 IPv4 地址

参数说明：

参数/返回值	类型	说明	数据
nettype	String	eth0:以太网 wlan0:wifi ppp0:电话卡	如 "eth0"
返回值	String	获取到的 IPV4 地址	

示例：

```
String ipAddress= baijieManager.getIPv4((String) (msg.obj));  
//ipAddress 为获取到的 IPV4 地址。
```

3.2 判断 wifi 是否打开

函数：public boolean isWifiEnabled();

描述：获取到 wifi 的状态是否打开；

参数说明：

参数/返回值	类型	说明	数据
返回值	boolean	false: 关闭 true: 打开	

示例：

```
boolean isWifiEnabled = baijieManager.isWifiEnabled();  
//判断当前 wifi 是否打开，返回 true 表示打开，false 表示关闭。
```

3.3 获取当前连接 wifi 名称

函数：public String getConnectWifiSsid();

描述：获取当前连接的 wifi 名称并去掉双引号；

参数说明：

参数/返回值	类型	说明	数据
--------	----	----	----

返回值	String	获取到连接的 wifi 名称, 未连接时为"unknown ssid"	
-----	--------	-------------------------------------	--

示例:

```
String Ssid = baijieManager.getConnectWifiSsid();  
//获取到已连接的 wifi 名称, 未连接时返回"unknown ssid"。
```

3.4 获取 wifi 列表信息

函数: public StringBuilder getWifiSsidInfo();

描述: 获取 wifi 列表信息;

参数说明:

参数/返回值	类型	说明	数据
返回值	StringBuilder	获取到 wifi 列表信息	

示例:

```
StringBuilder wifiListBuf = baijieManager.getWifiSsidInfo();  
//获取到 wifi 列表信息, 包括 SSID, BSSID, frequency 等。
```

3.5 设置当前连接 WIFI

函数: public boolean setWifi(Activity activity,String ssid, String password);

描述: 设置当前 wifi

参数说明:

参数/返回值	类型	说明	数据
activity	Activity	当前活动	
Ssid	String	wifi 账号	
Password	String	wifi 密码	
返回值	boolean	是否正确设置	

示例:

```
boolean result =  
baijieManager.setWifi(MainActivity.this, " baijiekeji" , " 12345678" );
```

3.6 设置网络优先级

函数：public void priorityScore(String ethScore , String wifiScore , String simScore);

描述：设置当前以太网，wifi，4g 网络优先级，分值越高优先级越大，最高为 100，以太网分值默认为 70，wifi 分值默认为 60,4g 网络默认为 50

参数说明：

参数/返回值	类型	说明	数据
ethScore	String	以太网分值	
wifiScore	String	Wifi 分值	
simScore	String	4g 网络分值	

示例：

```
baijieManager.priorityScore( "70" , " 60" , " 50" );
```

4 时间

4.1 获取系统日期

函数：public String getSystemDate();

描述：获取 Android 设备系统日期

参数说明：

参数/返回值	类型	说明	数据
返回值	String	获取 Android 系统日期，格式为 yyyy/MM/dd	

示例：

```
String systemDate = baijieManager.getSystemDate();  
//systemDate 为获取到的 Android 系统日期。
```

4.2 获取系统时间

函数：public String getSystemTime();

描述：获取 Android 设备系统时间

参数说明：

参数/返回值	类型	说明	数据
返回值	String	获取 Android 系统日期，格式为 hh:mm:ss	

示例：

```
String systemTime = baijieManager.getSystemTime();  
//systemTime 为获取到的 Android 系统时间
```

4.3 设置系统时间

函数：public void setSystemTime(String value);

描述：设置 Android 设备系统日期与时间。

说明：只需要传入 value 的设置时间即可。

参数说明：

参数/返回值	类型	说明	数据
String	String	设置日期时间	"\"2025-06-10 12:20:21\""

示例:

```
baijieManager.setSystemTime("\"2025-06-10 12:20:21\"");  
//设置 Android 设备系统日期为 2025 年 6 月 10 日 12 时 20 分 21 秒。
```

4.4 判断系统的时间是否自动获取

函数: public boolean isAutoDateTime();

描述: 判断 Android 系统的时间是否为自动获取

参数说明:

参数/返回值	类型	说明	数据
返回值	boolean	true:自动获取 false: 手动获取	true

示例:

```
boolean isAutoDateTime = baijieManager.isAutoDateTime();  
//isAutoDateTime 为系统的时间状态。
```

5 显示

5.1 获取 Android 系统的屏幕分辨率

函数：public String getDisplayMode();

描述：获取 Android 系统的屏幕分辨率

参数说明：

参数/返回值	类型	说明	数据
返回值	String	Android 系统的屏幕分辨率	1024x600

示例：

```
String displayMode = baijieManager.getDisplayMode();  
//displayMode 为获取到的 Android 系统的屏幕分辨率
```

5.2 获取 Android 设备显示分辨率 Y 轴像素

函数：public int getDisplayHeight();

描述：获取 Android 设备显示分辨率 Y 轴像素

参数说明：

参数/返回值	类型	说明	数据
返回值	int	分辨率 Y 轴像素	1024

示例：

```
int displayHeight = baijieManager.getDisplayHeight();  
//displayHeight 为获取到的 Android 系统的分辨率 Y 轴像素
```

5.3 获取 Android 设备显示分辨率 X 轴像素

函数：public int getDisplayWidth();

描述：获取 Android 设备显示分辨率 X 轴像素

参数说明：

参数/返回值	类型	说明	数据
返回值	int	分辨率 X 轴像素	600

示例：

```
int displayWidth= baijieManager.getDisplayWidth();  
//displayWidth 为获取到的 Android 系统的分辨率 X 轴像素
```

5.4 获取屏幕显示密度

函数: `public int getDisplayDensity();`

描述: 获取 Android 设备屏幕显示密度(dpi)

参数说明:

参数/返回值	类型	说明	数据
返回值	int	显示密度(dpi)	180

示例:

```
int displayDensity = baijieManager.getDisplayDensity();  
//displayDensity 为获取到的 Android 设备屏幕显示密度(dpi)
```

5.5 设置屏幕显示密度

函数: `public void setDisplayDensity(int dpis);`

描述: 设置 Android 设备屏幕显示密度(dpi)

参数说明:

参数/返回值	类型	说明	数据
dpis	int	0: 设置为 120, 1: 设置为 160, 2: 设置为 240, 3: 设置为 320. 或者直接填写显示密度值: 180,200 等	

示例:

```
baijieManager.setDisplayDensity(180);  
//设置 Android 设备屏幕显示密度(dpi)为 180
```

5.6 获取最大亮度值

函数: `public int getSystemMaxBrightness();`

描述: 获取 Android 设备系统最大亮度值

参数说明:

参数/返回值	类型	说明	数据
--------	----	----	----

返回值	int	系统最大亮度值	255
-----	-----	---------	-----

示例:

```
int systemMaxBrightness = baijieManager.getSystemMaxBrightness();  
//systemMaxBrightness 为获取到的 Android 设备最大亮度值。
```

5.7 增大亮度值

函数: public void increaseBrightness();

描述: 设置 Android 设备系统亮度值+1

示例:

```
baijieManager.increaseBrightness();  
//设置 Android 设备亮度值+1。
```

5.8 降低亮度值

函数: public void decreaseBrightness();

描述: 设置 Android 设备系统亮度值-1。

示例:

```
baijieManager.decreaseBrightness();  
//设置 Android 设备亮度值-1。
```

5.9 获取系统当前亮度值

函数: public int getSystemBrightness();

描述: 获取 Android 设备系统当前亮度值

参数说明:

参数/返回值	类型	说明	数据
返回值	int	系统亮度值	200

示例:

```
int systemBrightness = baijieManager.getSystemBrightness();  
//systemBrightness 为获取到的 Android 设备当前亮度值。
```

5.10 设置系统亮度值

函数: `public void setBrightness(int brightness);`

描述: 设置 Android 设备系统亮度值

参数说明:

参数/返回值	类型	说明	数据
brightness	int	系统亮度值	0-255

示例:

```
baijieManager.setBrightness(200);  
//设置 Android 设备亮度值为 200。
```

6 关于 root

6.1 整机 root

函数: `public void getRoot();`

描述: 设置整机 root

示例:

```
baijieManager.getRoot();
```

注意: android 13 需要运行两次, 第一次运行会重启, 重启后再运行一次