

Table of Contents

关于Titan	1.1
版本履历	1.2
1.0.0	1.2.1
开发者指南	1.3
鉴权方式	1.4
API 文档	1.5
API 错误码	1.5.1
titan.android	1.5.2
TitanContext	1.5.2.1
DeviceNotSupportedException	1.5.2.2
MethodNotSupportedException	1.5.2.3
NotPermissionException	1.5.2.4
RemoteExecuteException	1.5.2.5
ServiceDeathException	1.5.2.6
ServiceNotFoundException	1.5.2.7
titan.android.dataacq	1.5.3
DataAcqService	1.5.3.1
ComponentStatus	1.5.3.2
titan.android.dataacq.hardware	1.5.4
AudioJack	1.5.4.1
Battery	1.5.4.2
Bluetooth	1.5.4.3
Camera	1.5.4.4
CPU	1.5.4.5
EMMC	1.5.4.6
Ethernet	1.5.4.7
Flashlight	1.5.4.8
GMChip	1.5.4.9
ICCard	1.5.4.10
Key	1.5.4.11
LCD	1.5.4.12
Magcard	1.5.4.13
Modem	1.5.4.14
Pinpad	1.5.4.15
Printer	1.5.4.16

RFCard	1.5.4.17
SafeModel	1.5.4.18
Sensors	1.5.4.19
TouchPanel	1.5.4.20
USB	1.5.4.21
Wifi	1.5.4.22
titan.android.dataacq.software	1.5.5
Security	1.5.5.1
SystemUsage	1.5.5.2
titan.android.easytouch	1.5.6
EasyTouchService	1.5.6.1
ETShortcutInfo	1.5.6.2
titan.android.ethernet	1.5.7
EthernetService	1.5.7.1
DHCPInfo	1.5.7.2
titan.android.resetfactory	1.5.8
ResetFactoryService	1.5.8.1
titan.android.systemfunction	1.5.9
SystemFunctionService	1.5.9.1
titan.android.systemfunction.application	1.5.10
ApplicationModule	1.5.10.1
AppSignInfo	1.5.10.2
AppSizeInfo	1.5.10.3
UsageStatsInfo	1.5.10.4
titan.android.systemfunction.key	1.5.10.4.1
KeyModule	1.5.10.5
titan.android.systemfunction.location	1.5.11
LocationModule	1.5.11.1
titan.android.systemfunction.mount	1.5.12
MountModule	1.5.12.1
titan.android.systemfunction.nfc	1.5.13
NfcModule	1.5.13.1
titan.android.systemfunction.process	1.5.14
ProcessModule	1.5.14.1
titan.android.systemfunction.setting	1.5.15
SettingModule	1.5.15.1
IMEInfo	1.5.15.2
LanguageInfo	1.5.15.3
TimeZoneInfo	1.5.15.4

titan.android.systemfunction.statusbar	1.5.16
StatusBarModule	1.5.16.1
titan.android.systemfunction.telephony	1.5.17
TelephonyModule	1.5.17.1
titan.android.systemfunction.usb	1.5.18
UsbModule	1.5.18.1

系统统一接口

项目背景

AND平台的智能POS按不同客户需求分别定制不同的系统API提供给客户调用。这部分API以不同形式存在，如广播，JAR包，AIDL等不同方式提供给客户。此外，提供出去的这些系统API忽略了应用权限检查，使不同身份的应用都可以调用系统接口，因此存在极大的安全隐患。智能POS系统统一接口服务是为了提供一套统一的，规范化的，以及基于应用身份检查的系统API接口。智能POS系统统一接口服务整合当前已存在的定制化API，并将之按功能分类，统一分配权限，开发者使用同一套API，而无须关心设备及版本差异。

系统统一接口服务框架采用Android接口定义语言（AIDL）作为服务进程与客户进程交互的设计语言。接口服务按接口类型划分为几个独立模块，各个模块提供同一类系统功能的接口。如数据采集模块，应用包模块，设置模块，位置服务模块等。

适用平台

系统统一接口服务框架适用于AND平台，如APOS A8， C10等运行于Android5.0及以上联迪智能POS终端。

版本

1.0.0

发布日期

2018-10-30

版本号

1.0.0

发布日期

2018-10-30

主要更新

无

OS版本

• APOS

4.3.36

• C10

3.15.69

依赖模块版本

• APOS平台

titan_conf	1.0.0
titan_seccheck	1.0.0
titan_sysfunc	1.0.0
facreset	1.2.15a

• C10平台

titan_conf	1.0.0
titan_seccheck	1.0.0
titan_sysfunc	1.0.0
facreset	1.2.15a
easytouch	1.3.7

下载地址

[titanapi_1.0.0](#)

开发者指南

Titan整体架构



如上所示，Titan框架主要分由以下几个层次构成：

- SDK
- Service
- Android OS

SDK

即titanapi.jar, 包含了定制系统为上层调用提供的接口，开发应用程序时需要把titanapi.jar导入到工程里。

Service

即各个系统服务，如SystemFunctionService, DataAcqService等底层系统服务。这些服务一般由系统预置，开机启动。开发者需要关注的是titanapi.jar与各系统服务的版本依赖关系。这些关系可以在[版本](#)中找到对应的最低依赖关系。当版本不满足时，调用api会抛出异常或返回错误码。

Android OS

在原生系统上定制了厂商功能的基础运行环境。

鉴权方式

目前系统统一接口服务为每一个提供给开发者的API都定义了安全等级：

NORMAL：普通权限等级，不要求应用具备任何权限均可以调用，这部分API主要用于查询不敏感的系统功能，如系统语言，输入法，系统时间等。

MANAGER：管理权限等级，要求应用具备特定权限才可以调用，这部分API涉及系统安全，稳定性，改变系统属性等敏感操作，如设置系统语言，更新开机动画等。

调用MANAGER等级及以上API，要求应用必须通过以下某种鉴权方式，如果所有鉴权方式都不通过，则应用调用API会抛出安全异常。

使用系统统一接口服务框架，应用调用管理等级权限的API时，要求应用具有能够表明应用合法性的身份证明，如应用签名方式。接口服务框架支持应用三种安全检查机制。

- 扩展权限鉴权
- 签名证书鉴权
- API密钥鉴权

扩展权限鉴权

扩展权限是对于每个接口的访问权限进行单独定义。在对APK进行签名的过程中，会将已授权的权限打在该APK的签名信息中。在APK安装成功之后，相关的扩展权限信息会被提取并妥善保存。当该应用要访问相关的接口时，对应的底层服务，会检查调用者是否拥有所需要的权限。

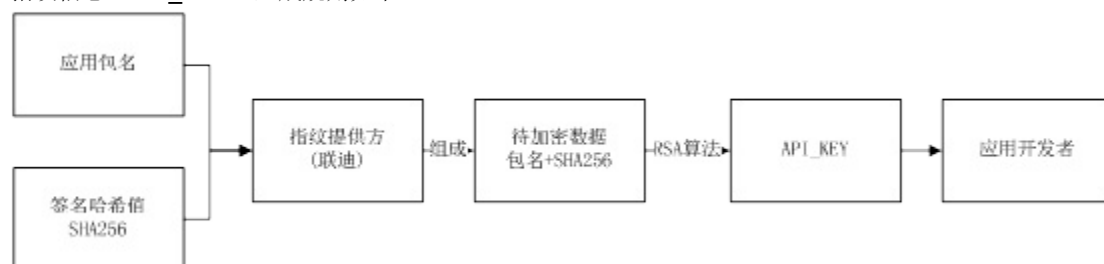
签名证书鉴权

签名证书鉴权指的应用使用联迪签名方案，联迪分配签名证书给应用签名，系统服务通过APK签名者信息来识别应用身份。系统服务要求应用签名者的身份等级不小于API的安全等级，API权限检查才能通过。即，某一API安全等级为MANAGER（权值为2），那么要求应用签名者身份等级要高于或等于MANAGER等级。

API密钥鉴权

API密钥鉴权指的应用不使用联迪签名方案，但需要向联迪申请应用的身份密钥，密钥使用非对称加密算法（RSA算）对应用的身份信息进行加密，生成一串指纹数据（API_KEY），系统服务检测应用API_KEY。通过身份检查，应用即可调用接口服务提供的所有API。

指纹信息（API_KEY）生成规则如下：



第三方应用提供应用包名以及签名哈希值SHA256（从keystore提取），联迪将包名跟SHA256通过联迪统一接口私钥加密生成一串指纹信息（API_KEY），开发者将指纹信息在应用AndroidManifest.xml中声明,如下：

```

<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <meta-data
        android:name="com.smartpos.API_KEY"
        android:value="58634A344778707569304D764661522B5A504E5059334A776354367A784B7531516A77..."
    />
    <activity android:name="com.smartpos.demo.MainActivity" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

```


Errors

since 1.0.0

概览

系统统一接口返回值常量定义类。调用任何一个API，都可以通过TitanContext.getLastError()接口获取对应的错误码。

常量

名称	取值	说明
NONE	0x00	成功
NOTPERMISSION	0x10001	应用无权限
METHOD_NOT_SUPPORTED	0x10002	系统不支持, 一般是由于系统软件功能不存在或版本过低
DEVICE_NOT_SUPPORTED	0x10003	设备不支持, 一般是由于终端不存在某一硬件模块, 如电池, NFC芯片等硬件模块
ERRPARAM	0x11001	参数错误
APP_NOT_RUNNING	0x21001	App未运行
APP_NOT_INSTALLED	0x21002	App未安装
APK_NOT_FOUND	0x21003	Apk文件未找到
APP_UNSIGNED	0x21004	App未签名
BOOT_ANIMATION_ZIP_NOT_EXIST	0x22001	开机动画zip包不存在
COPY_BOOT_ANIMATION_ZIP_FAIL	0x22002	拷贝开机动画zip包到指定路径失败
CHMOD_BOOT_ANIMATION_ZIP_FIAL	0x22003	修改开机动画文件权限失败
CURRENT_IME_IS_NULL	0x22004	未设置当前输入法
IME_NOT_INSTALLED	0x22005	输入法未安装
ERR_SIM_STATE	0x23001	SIM卡状态错误
SET_PREFERRED_NETWORK_TYPE_FAIL	0x23002	设置网络首选项失败
UNKNOWN_PREFERRED_NETWORK_TYPE	0x23003	未知类型的网络首选项

包名

titan.android

模块

- [TitanContext](#)
- [DeviceNotSupportedException](#)
- [MethodNotSupportedException](#)
- [NotPermissionException](#)
- [RemoteExecuteException](#)
- [ServiceDeathException](#)
- [ServiceNotFoundException](#)

TitanContext

Singleton Pattern

since 1.0.0

概览

系统统一接口上下文，应用程序使用系统统一接口服务的全局接口。通过TitanContext可以访问：

- [DataAcqService](#) 数据采集
- [EasyTouchService](#) 悬浮球
- [EthernetService](#) 以太网
- [ResetFactoryService](#) 恢复出厂设置
- [SystemFunctionService](#) 系统功能

等系统功能。

方法说明

构造方法

单例模式，不支持应用直接实例化，而是通过getContext(Context)接口获取实例。

getTitanContext

```
public static TitanContext getTitanContext(Context context)
```

功能

获取TitanContext实例。

参数

- context: 应用上下文。

返回值

TitanContext唯一实例。

getService

```
public T getService(@NonNull Class serviceType) throws ServiceNotFoundException
```

功能

获取统一接口远程服务在应用端的代理对象,通过服务代理对象访问系统功能。

参数

- **serviceType**: 服务代理对象类型，不可为null。枚举类型有：
 - DataAcqService.class
 - EasyTouchService.class
 - EthernetService.class
 - ResetFactoryService.class
 - SystemFunctionService.class

返回值

服务代理对象，由入参**serviceType**指定代理对象类型。

异常

- **ServiceNotFoundException**: **serviceType**所指类型服务不存在时抛出此异常。

编程示例:

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
    SystemFunctionService service = titanContext.getService(SystemFunctionService.class);
    ApplicationModule application = service.getFunction(ApplicationModule.class);
    AppSignInInfo appSignInInfo = application.getAppSignInInfo("com.example.demo");
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```


DeviceNotSupportedException

extends `titan.android.RemoteExecuteException`

since 1.0.0

概览

应用调用统一API时，设备不支持此功能时，抛出此异常，属于硬件范畴。如ACER设备不支持电源数据采集，当应用采集电源数据时，抛出此异常。

错误码

0x10003

MethodNotSupportedException

extends titan.android.RemoteExecuteException

since 1.0.0

概览

应用调用统一API时，统一接口服务不支持此功能时，抛出此异常, 属于软件系统范畴。

错误码

0x10002

NotPermissionException

extends titan.android.RemoteExecuteException

since 1.0.0

概览

应用无权限异常.

错误码

0x10001

RemoteExecuteException

extends `java.lang.Exception`

since 1.0.0

概览

系统统一接口功能执行异常。

应用使用统一API接口时，当出现以下几种情况时会抛出RemoteExecuteException异常或其子类异常：

- 参数错误时抛出 RemoteExecuteException。
- 应用无权限时抛出 NotPermissionException。
- 设备不支持,如设备不具备执行此接口的条件时抛出 DeviceNotSupportedException。
- 功能不支持,如系统服务未升级到满足该接口的版本或与远程服务交互异常时抛出 MethodNotSupportedException。

子类

- [DeviceNotSupportedException](#)
- [MethodNotSupportedException](#)
- [NotPermissionException](#)

方法说明

getErrorCode

`public int getErrorCode()`

功能

获取异常错误码。

返回值

错误码。参阅 [错误码](#)。

getLocalizedMessage

`public String getLocalizedMessage()`

功能

获取异常信息。

返回值

异常信息，包含错误码。参阅 [错误码](#)。

ServiceDeathException

extends `java.lang.Exception`

since 1.0.0

概览

应用调用统一接口时，接口服务未运行时抛出此异常.服务未运行一般有以下情况:

- 服务未安装
- 服务已安装但未启动
- 服务运行过程中出现崩溃，崩溃后未重启

ServiceNotFoundException

extends `java.lang.Exception`

since 1.0.0

概览

当应用调用 `TitanContext.getService(Class)` 传入的 `class` 类型不被识别时抛出此异常。

包名

titan.android.dataacq

模块

- [数据采集服务](#)
- [采集模块状态量](#)

DataAcqService

extends titan.android.core.TitanService

since 1.0.0

概览

系统统一接口数据采集服务. 用于获取设备的统计量和状态量. 不能直接实例化DataAcqService, 而是通过TitanContext.getService(Class)接口, 传入参数 DataAcqService.class 获取采集服务代理对象.

需要采集某一模块的统计量或状态量时, 通过getComponent(Class) 接口获取对应模块的实例, 入参为模块的类型, 如获取音频孔模块, 传入AudioJack.class, 如果远程采集服务未运行, 则抛出 ServiceDeathException异常.

方法说明

构造方法

不支持应用直接实例化, 而是通过TitanContext.getService(DataAcqService.class)接口获取实例。

getComponent

```
public T getComponent(@NonNull Class clazz) throws ServiceDeathException, ServiceNotFoundException
```

功能

获取采集模块。

参数

- **clazz**: 服务代理对象类型, 不可为null。枚举类型有:
 - AudioJack.class
 - Battery.class
 - Bluetooth.class
 - Camera.class
 - CPU.class
 - 其他类型...

返回值

采集模块实例, 实例类型即为入参clazz指定的类型。

异常

- ServiceNotFoundException: clazz所指模块类型不存在时抛出此异常。
- ServiceDeathException: 远程服务未运行时抛出

编程示例:

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    AudioJack audioJack = dataAcq.getComponent(AudioJack.class);
    // 获取音频头插入次数统计量
    long count = audioJack.getUsage(AudioJack.PLUG_COUNT);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
}
```

ComponentStatus

since 1.0.0

概览

采集模块状态量包装类。

枚举类

Level

状态码等级

- S：正常
- E：故障
- W：警告

方法说明

getEncoder

```
public String getEncoder()
```

功能

获取编码定义方。

返回值

编码定义方。

getLevel

```
public Level getLevel()
```

功能

获取状态码级别。

返回值

状态码级别。

getDescription

```
public String getDescription()
```

功能

获取状态码简单描述。

返回值

状态码简单描述。

| getTagNo

```
public int getTagNo()
```

功能

获取采集因子tagNo。

返回值

采集因子tagNo。

| getFactorNo

```
public int getFactorNo()
```

功能

获取采集因子factorNo。

返回值

采集因子factorNo。

| getStatusCode

```
public String getStatusCode()
```

功能

获取状态码，16进制表示的状态编码，如0x41。

返回值

状态码。

包名

titan.android.dataacq.hardware

模块

- 音频孔数据采集
- 电池数据采集
- 蓝牙数据采集
- 摄像头数据采集
- CPU数据采集
- EMMC模块数据采集
- 以太网数据采集
- 闪光灯数据采集
- 国密芯片数据采集
- 接触式卡模块数据采集
- 按键数据采集
- 显示屏数据采集
- 磁条卡模块数据采集
- Modem数据采集
- Pinpad模块数据采集
- 打印设备数据采集
- 非接触式射频模块数据采集
- 安全模块数据采集
- 传感器数据采集
- 触屏数据采集
- USB数据采集
- Wifi数据采集

AudioJack

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

音频孔数据采集. 通过DataAcqService.getComponent(AudioJack.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
PLUG_COUNT	Long	获取音频头插入次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(AudioJack.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - PLUG_COUNT

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    AudioJack audioJack = dataAcq.getComponent(AudioJack.class);
    //获取音频头插入次数
    long count = audioJack.getUsage(AudioJack.PLUG_COUNT);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

Battery

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

电池数据采集. 通过DataAcqService.getComponent(Battery.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
POWER_LEVEL	String	获取电池电量
BATTERY_VOLTAGE	String	获取电池电压
STANDBY_BATTERY_VOLTAGE	String	获取备份电池电压
CHARGE_DURATION	Long	获取充电累计时间(分钟)
POWER_USAGE	String	获取电量使用情况(mAh)
CHARGE_COUNT	Long	获取充电次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Battery.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入[采集常量](#):
 - POWER_LEVEL
 - BATTERY_VOLTAGE
 - STANDBY_BATTERY_VOLTAGE
 - CHARGE_DURATION
 - POWER_USAGE

- CHARGE_COUNT

返回值

统计量。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getStatus

`public ComponentStatus getStatus() throws ServiceDeathException, RemoteExecuteException`

功能

获取电池状态量信息。

返回值

电池状态量信息。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    Battery battery = dataAcq.getComponent(Battery.class);
    //获取充电累计时间(分钟)
    long count = battery.getUsage(Battery.CHARGE_DURATION);
    //获取电池电压
    String voltage = battery.getUsage(Battery.BATTERY_VOLTAGE);
    //获取电池状态
    ComponentStatus status = battery.getStatus();
    //状态码等级
    Level level = status.getLevel();
    //状态码
    String statusCode = status.getStatusCode();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
}
```

```
} catch (RemoteExecuteException e) {  
    e.printStackTrace();  
}
```

Bluetooth

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

蓝牙数据采集. 通过DataAcqService.getComponent(Bluetooth.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
BLUETOOTH_MODEL	String	获取蓝牙型号
BLUETOOTH_MAC	String	获取蓝牙MAC地址
BLUETOOTH_OPEN_COUNT	Long	获取蓝牙打开次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Bluetooth.class)接口获取实例。

getUsage

```
public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException
```

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - BLUETOOTH_MODEL
 - BLUETOOTH_MAC
 - BLUETOOTH_OPEN_COUNT

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。

- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getStatus

public ComponentStatus getStatus() throws ServiceDeathException, RemoteExecuteException

功能

获取蓝牙运行状态。

返回值

蓝牙运行状态量信息。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    Bluetooth bluetooth = dataAcq.getComponent(Bluetooth.class);
    //获取蓝牙型号
    String model = bluetooth.getUsage(Bluetooth.BLUETOOTH_MODEL);
    //获取蓝牙运行状态
    ComponentStatus status = bluetooth.getStatus();
    //状态码等级
    Level level = status.getLevel();
    //状态码
    String statusCode = status.getStatusCode();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

Camera

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

摄像头数据采集. 通过DataAcqService.getComponent(Camera.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
FRONT_CAMERA_COUNT	Long	获取前置摄像头打开次数
BACK_CAMERA_COUNT	Long	获取后置摄像头打开次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Camera.class)接口获取实例。

getUsage

```
public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException
```

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - FRONT_CAMERA_COUNT
 - BACK_CAMERA_COUNT

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限

- 数据解析失败
- 采集因子不被支持
- 系统不支持

getFrontCameraStatus

`public ComponentStatus getFrontCameraStatus() throws ServiceDeathException, RemoteExecuteException`

功能

获取前置摄像头状态。

返回值

前置摄像头状态。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getBackCameraStatus

`public ComponentStatus getBackCameraStatus() throws ServiceDeathException, RemoteExecuteException`

功能

获取后置摄像头状态。

返回值

后置摄像头状态。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    Camera camera = dataAcq.getComponent(Camera.class);
    //获取前置摄像头打开次数
    long count = camera.getUsage(Camera.FRONT_CAMERA_COUNT);
}
```

```
// 获取前置摄像头状态
ComponentStatus status = camera.getFrontCameraStatus();
//状态码等级
Level level = status.getLevel();
//状态码
String statusCode = status.getStatusCode();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

CPU

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

CPU数据采集. 通过DataAcqService.getComponent(CPU.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
CPU_MODEL	String	获取CPU型号

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(CPU.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- **request**: 采集请求，不可为null。传入[采集常量](#):
 - CPU_MODEL

返回值

统计量。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    CPU cpu = dataAcq.getComponent(CPU.class);
    //获取CPU型号
    String model = cpu.getUsage(CPU.CPU_MODEL);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

EMMC

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

EMMC模块采集. 通过DataAcqService.getComponent(EMMC.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
EMMC_MANUFACTURER	String	获取EMMC厂商
EMMC_MODEL	String	获取EMMC型号
EMMC_LIFE	String	获取EMMC使用寿命

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(EMMC.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - EMMC_MANUFACTURER
 - EMMC_MODEL
 - EMMC_LIFE

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。

- RemoteExecuteException 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    EMMC emmc = dataAcq.getComponent(EMMC.class);
    //获取EMMC厂商
    String manu = emmc.getUsage(EMMC.EMMC_MANUFACTURER);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

Flashlight

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

闪光灯数据采集. 通过DataAcqService.getComponent(Flashlight.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
FLASH_OPEN_COUNT	Long	获取闪光灯打开次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Flashlight.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - FLASH_OPEN_COUNT

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(Flashlight.class);
    Flashlight flashlight = dataAcq.getComponent(Flashlight.class);
    //获取闪光灯打开次数
    Long count = flashlight.getUsage(Flashlight.FLASH_OPEN_COUNT);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

GMChip

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

国密芯片数据采集. 通过DataAcqService.getComponent(GMChip.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
ENCRYPT_IC_MODEL	String	获取国密芯片型号

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(GMChip.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - ENCRYPT_IC_MODEL

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    GMChip chip = dataAcq.getComponent(GMChip.class);
    //获取国密芯片型号
    String model = chip.getUsage(GMChip.ENCRYPT_IC_MODEL);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

ICCard

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

接触式卡模块数据采集. 通过DataAcqService.getComponent(ICCard.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
POWERUP_SUCCESS_COUNT	Long	获取IC卡上电成功次数
POWERUP_FAIL_COUNT	Long	获取IC卡上电失败次数
APDU_FAIL_COUNT	Long	获取IC卡APDU失败次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(ICCard.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - POWERUP_SUCCESS_COUNT
 - POWERUP_FAIL_COUNT
 - APDU_FAIL_COUNT

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。

- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getICCard1Status

public ComponentStatus getICCard1Status() throws ServiceDeathException, RemoteExecuteException

功能

获取IC卡设备状态。

返回值

IC卡设备状态。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getICCard2Status

public ComponentStatus getICCard2Status() throws ServiceDeathException, RemoteExecuteException

功能

获取IC卡2设备状态。

返回值

IC卡2设备状态。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getSAM1CardStatus

public ComponentStatus getSAM1CardStatus() throws ServiceDeathException, RemoteExecuteException

功能

获取SAM1设备状态。

返回值

SAM1设备状态。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getSAM2CardStatus

`public ComponentStatus getSAM2CardStatus() throws ServiceDeathException, RemoteExecuteException`

功能

获取SAM2设备状态。

返回值

SAM2设备状态。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getSAM3CardStatus

`public ComponentStatus getSAM3CardStatus() throws ServiceDeathException, RemoteExecuteException`

功能

获取SAM3设备状态。

返回值

SAM3设备状态。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持

- 系统不支持

getSAM4CardStatus

public ComponentStatus getSAM4CardStatus() throws ServiceDeathException, RemoteExecuteException

功能

获取SAM4设备状态。

返回值

SAM4设备状态。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    ICCard iccard = dataAcq.getComponent(ICCard.class);
    //获取IC卡上电成功次数
    long count = iccard.getUsage(ICCard.POWERUP_SUCCESS_COUNT);
    //获取ICCard设备1状态
    ComponentStatus status = iccard.getICCard1Status();
    //状态码等级
    Level level = status.getLevel();
    //状态码
    String statusCode = status.getStatusCode();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

Key

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

按键数据采集. 通过DataAcqService.getComponent(Key.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
POWER_KEY_CLICK_COUNT	Long	获取电源键按键次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Key.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - POWER_KEY_CLICK_COUNT

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    Key key = dataAcq.getComponent(Key.class);
    // 获取电源键按键次数
    long count = key.getUsage(Key.POWER_KEY_CLICK_COUNT);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

LCD

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

显示屏数据采集. 通过DataAcqService.getComponent(LCD.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
SCREENON_DURATION	Long	获取屏幕背光点亮时间(分钟)
SCREEN_CLICK_COUNT	Long	获取屏幕点击次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(LCD.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - SCREENON_DURATION
 - SCREEN_CLICK_COUNT

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时

- 数据解析失败
- 采集因子不被支持
- 系统不支持

getStatus

`public ComponentStatus getStatus() throws ServiceDeathException, RemoteExecuteException`

功能

获取显示屏接口通信状态。

返回值

显示屏接口通信状态。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    LCD lcd = dataAcq.getComponent(LCD.class);
    // 获取屏幕点击次数
    long count = lcd.getUsage(LCD.SCREEN_CLICK_COUNT);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

Magcard

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

磁条卡模块数据采集. 通过DataAcqService.getComponent(Magcard.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
SWIPE_COUNT	Long	获取刷卡总次数
TRK1_SUCCESS_COUNT	Long	获取1磁道成功次数
TRK1_FAIL_COUNT	Long	获取1磁道失败次数
TRK2_FAIL_COUNT	Long	获取2磁道失败次数
TRK3_SUCCESS_COUNT	Long	获取3磁道成功次数
TRK3_FAIL_COUNT	Long	获取3磁道失败次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Battery.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入[采集常量](#):
 - SWIPE_COUNT
 - TRK1_SUCCESS_COUNT
 - TRK1_FAIL_COUNT
 - TRK2_FAIL_COUNT
 - TRK3_SUCCESS_COUNT

- TRK3_FAIL_COUNT

返回值

统计量。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getStatus

`public ComponentStatus getStatus() throws ServiceDeathException, RemoteExecuteException`

功能

获取磁卡模块状态。

返回值

电池状态量信息。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    Magcard magcard = dataAcq.getComponent(Magcard.class);
    //获取刷卡总次数
    long count = magcard.getUsage(Magcard.SWIPE_COUNT);
    //获取磁卡模块状态
    ComponentStatus status = magcard.getStatus();
    //状态码等级
    Level level = status.getLevel();
    //状态码
    String statusCode = status.getStatusCode();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

}

Modem

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

Modem数据采集. 通过DataAcqService.getComponent(Modem.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
DIAL_COUNT	Long	获取MODEM拨号次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Modem.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - DIAL_COUNT

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    Modem modem = dataAcq.getComponent(Modem.class);
    //获取MODEM拨号次数
    long count = modem.getUsage(Modem.DIAL_COUNT);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

Pinpad

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

Pinpad模块数据采集. 通过DataAcqService.getComponent(Pinpad.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
OFFLINE_COUNT	Long	获取脱机PIN输入次数
ONLINE_COUNT	Long	获取联机PIN输入次数
NOPIN_COUNT	Long	获取非PIN输入次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Battery.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - OFFLINE_COUNT
 - ONLINE_COUNT
 - NOPIN_COUNT

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。

- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getStatus

public ComponentStatus getStatus() throws ServiceDeathException, RemoteExecuteException

功能

获取密码键盘状态。

返回值

密码键盘状态。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    Pinpad pinpad = dataAcq.getComponent(Pinpad.class);
    //获取联机PIN输入次数
    long count = pinpad.getUsage(Pinpad.ONLINE_COUNT);
    //获取密码键盘状态
    ComponentStatus status = pinpad.getStatus();
    //状态码等级
    Level level = status.getLevel();
    //状态码
    String statusCode = status.getStatusCode();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

Printer

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

打印设备数据采集. 通过DataAcqService.getComponent(Printer.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
HEAT_TEMP_COUNT	Long	获取打印高温次数
LOW_TEMP_COUNT	Long	获取打印低温次数
FEED_COUNT	Long	获取走纸电机步进量
PRINTER_MODEL	String	获取打印机芯型号
PRINTER_PAPER_LENGTH	Long	获取打印走纸长度(毫米)

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Printer.class)接口获取实例。

getUsage

```
public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException
```

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入[采集常量](#):
 - HEAT_TEMP_COUNT
 - LOW_TEMP_COUNT
 - FEED_COUNT
 - PRINTER_MODEL
 - PRINTER_PAPER_LENGTH

返回值

统计量。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getStatus

`public ComponentStatus getStatus() throws ServiceDeathException, RemoteExecuteException`

功能

获取打印设备状态。

返回值

打印设备状态。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    Printer printer = dataAcq.getComponent(Printer.class);
    //获取打印高温次数
    long count = printer.getUsage(Printer.HEAT_TEMP_COUNT);
    //获取打印设备状态
    ComponentStatus status = printer.getStatus();
    //状态码等级
    Level level = status.getLevel();
    //状态码
    String statusCode = status.getStatusCode();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```


RFCard

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

非接触式射频模块数据采集. 通过DataAcqService.getComponent(RFCard.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
ACTIVATE_SUCCESS_COUNT	Long	获取射频卡激活成功量
ACTIVATE_FAIL_COUNT	Long	获取射频卡激活失败量
APDU_FAIL_COUNT	Long	获取射频卡数据交换失败量
RFCARD_MODEL	String	获取射频卡芯片型号

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(RFCard.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入[采集常量](#):
 - ACTIVATE_SUCCESS_COUNT
 - ACTIVATE_FAIL_COUNT
 - APDU_FAIL_COUNT
 - RFCARD_MODEL

返回值

统计量。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getStatus

`public ComponentStatus getStatus() throws ServiceDeathException, RemoteExecuteException`

功能

获取射频卡设备状态。

返回值

射频卡设备状态。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    RFCard rfcard = dataAcq.getComponent(RFCard.class);
    //获取射频卡激活成功量
    long count = rfcard.getUsage(RFCard.ACTIVATE_SUCCESS_COUNT);
    //获取射频卡设备状态
    ComponentStatus status = rfcard.getStatus();
    //状态码等级
    Level level = status.getLevel();
    //状态码
    String statusCode = status.getStatusCode();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```


SafeModel

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

安全模块数据采集. 通过DataAcqService.getComponent(SafeModel.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
SAFE_MODEL	String	获取安全模块型号

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(SafeModel.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - SAFE_MODEL

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    SafeModel safeModel = dataAcq.getComponent(SafeModel.class);
    //获取安全模块型号
    String model = safeModel.getUsage(SafeModel.SAFE_MODEL);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

Sensors

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

传感器数据采集. 通过DataAcqService.getComponent(Sensors.class)获取实例。

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Sensors.class)接口获取实例。

getLightSensorStatus

public ComponentStatus getLightSensorStatus() throws ServiceDeathException, RemoteExecuteException

功能

获取光传感器通信状态。

返回值

光传感器通信状态。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getAccSensorStatus

public ComponentStatus getAccSensorStatus() throws ServiceDeathException, RemoteExecuteException

功能

获取加速传感器通信状态。

返回值

加速传感器通信状态。

异常

- `ServiceDeathException` 远程服务未运行时抛出。
- `RemoteExecuteException` 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    Sensors sensors = dataAcq.getComponent(Sensors.class);
    //获取打印设备状态
    ComponentStatus status = sensors.getLightSensorStatus();
    //状态码等级
    Level level = status.getLevel();
    //状态码
    String statusCode = status.getStatusCode();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```


TouchPanel

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

触屏数据采集. 通过DataAcqService.getComponent(TouchPanel.class)获取实例。

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(TouchPanel.class)接口获取实例。

getStatus

public ComponentStatus getStatus() throws ServiceDeathException, RemoteExecuteException

功能

获取触屏通信状态。

返回值

触屏通信状态。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    TouchPanel touchPanel = dataAcq.getComponent(TouchPanel.class);
    //获取触屏通信状态
    ComponentStatus status = touchPanel.getStatus();
    //状态码等级
    Level level = status.getLevel();
    //状态码
    String statusCode = status.getStatusCode();
}
```

```
} catch (ServiceNotFoundException e) {  
    e.printStackTrace();  
}  
} catch (ServiceDeathException e) {  
    e.printStackTrace();  
}  
} catch (RemoteExecuteException e) {  
    e.printStackTrace();  
}  
}
```

USB

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

USB数据采集. 通过DataAcqService.getComponent(USB.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
USB_PLUG_COUNT	Long	获取USB连接次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(USB.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - USB_PLUG_COUNT

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    USB usb = dataAcq.getComponent(USB.class);
    //获取USB连接次数
    long count = usb.getUsage(USB.USB_PLUG_COUNT);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

Wifi

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

Wifi数据采集. 通过DataAcqService.getComponent(Wifi.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
WIFI_MODEL	String	获取WIFI型号
WIFI_MAC	String	获取WIFI MAC地址

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Wifi.class)接口获取实例。

getUsage

```
public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException
```

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - WIFI_MODEL
 - WIFI_MAC

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时

- 数据解析失败
- 采集因子不被支持
- 系统不支持

getStatus

public ComponentStatus getStatus() throws ServiceDeathException, RemoteExecuteException

功能

获取WIFI驱动加载状态。

返回值

WIFI驱动加载状态。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    Wifi wifi = dataAcq.getComponent(Wifi.class);
    //获取WIFI MAC地址
    String mac = wifi.getUsage(Wifi.WIFI_MAC);
    //获取WIFI驱动加载状态
    ComponentStatus status = wifi.getStatus();
    //状态码等级
    Level level = status.getLevel();
    //状态码
    String statusCode = status.getStatusCode();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

包名

titan.android.dataacq.software

模块

- [安全模块数据采集](#)
- [系统信息数据采集](#)

Security

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

安全模块数据采集. 通过DataAcqService.getComponent(Security.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
CLEARATTACK_COUNT	Long	获取清除攻击次数

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(Security.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - CLEARATTACK_COUNT

返回值

统计量。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

getSecAttackDetectStatus

public ComponentStatus getSecAttackDetectStatus() throws ServiceDeathException, RemoteExecuteException

功能

获取SEC攻击检测状态。

返回值

SEC攻击检测状态。

异常

- ServiceDeathException 远程服务未运行时抛出。
- RemoteExecuteException 以下错误时抛出：
 - 应用无权限
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    Security security = dataAcq.getComponent(Security.class);
    //获取清除攻击次数
    long count = security.getUsage(Security.CLEARATTACK_COUNT);
    //获取SEC攻击检测
    ComponentStatus status = security.getSecAttackDetectStatus();
    //状态码等级
    Level level = status.getLevel();
    //状态码
    String statusCode = status.getStatusCode();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

SystemUsage

extends titan.android.core.dadaacq.StatsComponent

since 1.0.0

概览

系统信息数据采集. 通过DataAcqService.getComponent(SystemUsage.class)获取实例。

采集常量

名称	getUsage 返回类型	说明
POWERON_COUNT	Long	获取开机次数
SHUTDOWN_COUNT	Long	获取关机次数
LIFECYCLE_TOTAL	Long	获取累计运行时间(分钟)
PRODUCT_NAME	String	获取品牌名称
MODEL_NAME	String	获取型号
DEVICE_NAME	String	获取设备名称
SDCARD_TOTAL	Long	获取内置SD卡容量(bytes)
SDCARD_AVAILABLE	Long	获取内置SD卡可用大小(bytes)
NET_TYPE	String	获取网络类型
SDK_VERSION	String	获取SDK版本
ANDROID_VERSION	String	获取Android版本
UPTIME_INFO	Long	获取系统运行时间(分钟)
BASEBAND_VERSION	String	获取基带版本
SYSTEM_VERSION	String	获取系统版本
SMARTOS_VERSION	String	获取SmartOS版本
MEMORY_TOTAL	Long	获取系统总内存大小(bytes)
MEMORY_AVAILABLE	Long	获取系统可用内存大小(bytes)
DEVICE_MANUFACTURER	String	获取厂商名称
SERIAL_NO	String	获取序列号
IMEI	String	获取IMEI号
MEID	String	获取MEID号
SAFE_PATCH_VERSION	String	获取安全补丁版本

KERNEL_VERSION	String	获取内核版本号
ROM_TOTAL	Long	获取机身存储容量(bytes)
ROM_AVAILABLE	Long	获取机身存储可用大小(bytes)
TFCARD_TOTAL	Long	获取TF卡容量(bytes)
TFCARD_AVAILABLE	Long	获取TF卡可用大小(bytes)
IP_ADDRESS	String	获取IP地址
BOOT_VERSION	String	获取Boot版本
SDCARD_STATUS	String	获取内置SD卡状态
TFCARD_STATUS	String	获取TF卡状态

方法说明

构造方法

不支持应用直接实例化，而是通过DataAcqService.getComponent(SystemUsage.class)接口获取实例。

getUsage

public T getUsage(DataAcqRequest request) throws ServiceDeathException, RemoteExecuteException

功能

获取采集数据。

参数

- request: 采集请求，不可为null。传入采集常量：
 - POWERON_COUNT
 - SHUTDOWN_COUNT
 - LIFECYCLE_TOTAL
 - PRODUCT_NAME
 - MODEL_NAME
 - DEVICE_NAME
 - SDCARD_TOTAL
 - SDCARD_AVAILABLE
 - NET_TYPE
 - SDK_VERSION
 - ANDROID_VERSION
 - UPTIME_INFO
 - BASEBAND_VERSION
 - SYSTEM_VERSION
 - SMARTOS_VERSION
 - MEMORY_TOTAL
 - MEMORY_AVAILABLE
 - DEVICE_MANUFACTURER
 - SERIAL_NO

- IMEI
- MEID
- SAFE_PATCH_VERSION
- KERNEL_VERSION
- ROM_TOTAL
- ROM_AVAILABLE
- TFCARD_TOTAL
- TFCARD_AVAILABLE
- IP_ADDRESS
- BOOT_VERSION
- SDCARD_STATUS
- TFCARD_STATUS

返回值

统计量。

异常

- **ServiceDeathException** 远程服务未运行时抛出。
- **RemoteExecuteException** 以下错误时抛出：
 - 应用无权限时
 - 数据解析失败
 - 采集因子不被支持
 - 系统不支持

编程示例：

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    DataAcqService dataAcq = titan.getService(DataAcqService.class);
    SystemUsage systemUsage = dataAcq.getComponent(SystemUsage.class);
    //获取开机次数
    long count = systemUsage.getUsage(SystemUsage.POWERON_COUNT);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

包名

titan.android.easytouch

模块

- [悬浮球服务](#)
- [悬浮球快捷菜单配置信息](#)

EasyTouchService

extends titan.android.core.TitanService

since 1.0.0

概览

EasyTouchService悬浮球服务代理接口. 不能直接实例化EasyTouchService, 而是通过TitanContext.getService(Class)接口, 传入参数 EasyTouchService.class 获取悬浮球服务代理对象.

方法说明

构造方法

不支持应用直接实例化, 而是通过TitanContext.getService(EasyTouchService.class)接口获取实例。

addETShortcut

```
public boolean addETShortcut(@NonNull ETShortcutInfo info, boolean replace) throws ServiceDeathException
```

功能

添加悬浮快捷菜单。

参数

- **info**: 快捷菜单配置信息, 不可为null。
- **replace**: 当相同位置有其他应用快捷菜单时, 是否替换. true表示替换, false表示不替换。

返回值

true表示添加成功, false表示添加失败. 可通过TitanContext.getLastError() 查询错误码。

异常

- **ServiceDeathException**: 远程服务未运行时抛出

getETShortcuts

```
public List getETShortcuts() throws ServiceDeathException, RemoteExecuteException
```

功能

获取悬浮球已设置的快捷菜单信息。

返回值

快捷菜单信息列表。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

removeETShortcut

```
public boolean removeETShortcut(@NonNull ETShortcutInfo info) throws ServiceDeathException
```

功能

移除悬浮快捷菜单。

参数

- **info**: 快捷菜单配置信息, 不可为null。

返回值

true表示移除成功, **false**表示移除失败. 可通过**TitanContext.getLastError()** 查询错误码。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。

setEasyTouchEnabled

```
public boolean setEasyTouchEnabled(boolean enabled) throws ServiceDeathException
```

功能

开启/关闭悬浮球。

参数

- **enabled**: **true**表示开启, **false**表示关闭。

返回值

true表示移除成功, **false**表示移除失败. 可通过**TitanContext.getLastError()** 查询错误码。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。

setETShortcuts

```
public boolean setETShortcuts(@NonNull List infos) throws ServiceDeathException
```

功能

设置悬浮快捷菜单, 该接口将替换悬浮球上所有已设置的快捷菜单。

参数

- **infos**: 快捷菜单配置信息列表, 不可为null, 且列表至少有一项快捷菜单信息。

返回值

true表示设置成功, **false**表示设置失败. 可通过TitanContext.getLastError() 查询错误码。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。

turnplateSliding

public boolean turnplateSliding(boolean sliding) throws ServiceDeathException

功能

设置悬浮球转盘是否可转动。

参数

- **sliding**: **true**表示允许转盘转动, **false**表示禁止转移转动。

返回值

true表示设置成功, **false**表示设置失败. 可通过TitanContext.getLastError() 查询错误码。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。

编程示例：

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
    EasyTouchService service = titanContext.getService(EasyTouchService.class);
    List<ETShortcutsInfo> infos = service.getETShortcuts();
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```


ETShortcutInfo

implements android.os.Parcelable

since 1.0.0

概览

悬浮球快捷菜单配置信息。

公开属性

属性名	类型	说明
menuTitle	String	在悬浮球上显示的菜单名称
packageName	String	应用的包名
className	String	应用的入口组件类名, 类名为全路径
showIndex	int	应用显示在悬浮球上的位置, 以顺时针方向开始, 0表示位于悬浮球正上方
isLock	boolean	设置应用是否锁定, 锁定后的应用无法通过悬浮球的菜单设置进行移除

包名

titan.android.ethernet

模块

- 以太网服务
- DHCP元信息

EthernetService

extends titan.android.core.TitanService

since 1.0.0

概览

系统统一接口以太网服务. 用于查询以太网模块的设置信息. 不能直接实例化**EthernetService**, 而是通过TitanContext.getService(Class)接口, 传入参数 EthernetService.class 获取采集服务代理对象.

方法说明

构造方法

不支持应用直接实例化, 而是通过TitanContext.getService(EthernetService.class)接口获取实例。

isEthernetExist

public boolean isEthernetExist(String iface) throws ServiceDeathException, RemoteExecuteException

功能

检查以太网模块是否存在. 仅检查以太网硬件模块是否存在, 不涉及以太网网络是否连接。

参数

- **iface**: 以太网的物理接口名称, 一般为"eth0"。

返回值

存在返回true, 不存在返回false。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持
 - 设备不支持
 - 参数错误

setEthernetEnabled

public boolean setEthernetEnabled(boolean enabled) throws ServiceDeathException

功能

启用/禁用以太网模块. 仅设置以太网硬件模块, 不涉及以太网网络是否连接。

参数

- **enabled**: 启用:true, 禁用:false。

返回值

设置成功返回true,否则返回false。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**:接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 系统不支持
 - 设备不支持

isEthernetEnabled

`public boolean isEthernetEnabled() throws ServiceDeathException, RemoteExecuteException`

功能

检查以太网模块是否启用, 仅检查以太网硬件模块, 不涉及以太网网络是否连接。

返回值

启用返回true,禁用返回false。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**:接口执行过程异常, 以下情况会抛出异常
 - 系统不支持
 - 设备不支持

getEthernetInterfaceName

`public String getEthernetInterfaceName() throws ServiceDeathException, RemoteExecuteException`

功能

获得以太网的物理接口名称。

返回值

物理接口名称, 一般为"eth0"。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**:接口执行过程异常, 以下情况会抛出异常
 - 系统不支持
 - 设备不支持

getEthernetInterfaceState

```
public int getEthernetIfaceState() throws ServiceDeathException, RemoteExecuteException
```

功能

获得以太网的物理接口的状态。

返回值

物理接口状态值：

- 0：表示DOWN
- 1：表示UP

异常

- **ServiceDeathException**：远程服务未运行时抛出。
- **RemoteExecuteException**：接口执行过程异常，以下情况会抛出异常
 - 系统不支持
 - 设备不支持

| getEthernetCarrierState

```
public int getEthernetCarrierState() throws ServiceDeathException, RemoteExecuteException
```

功能

获得网线的连接状态。

返回值

当以太网的物理接口状态为UP的时候,即getEthernetIfaceState()返回值为1时,读取到的才是准确的,否则返回都是0,返回值有:

- 0:网线未插入
- 1:网线已连接

异常

- **ServiceDeathException**：远程服务未运行时抛出。
- **RemoteExecuteException**：接口执行过程异常，以下情况会抛出异常
 - 系统不支持
 - 设备不支持

| getEthernetHwaddr

```
public String getEthernetHwaddr(String iface) throws ServiceDeathException, RemoteExecuteException
```

功能

获得以太网的物理地址。

参数

- **iface**：以太网的物理接口名称,一般为"eth0"。

返回值

以太网物理地址。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持
 - 设备不支持
 - 参数错误

getDHCPInfo

`public DHCPInfo getDHCPInfo() throws ServiceDeathException, RemoteExecuteException`

功能

获得DHCP信息, 应在网络连接成功后调用该接口, 才能保证获取到正确的信息。

返回值

DHCP元信息。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持
 - 设备不支持

编程示例:

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    EthernetService service = titanContext.getService(EthernetService.class);
    boolean exist = service.isEthernetExist("eth0");
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

DHCPInfo

implements android.os.Parcelable

since 1.0.0

概览

DHCP元信息。

方法说明

getIpAddress

```
public String getIpAddress()
```

功能

获取IP地址。

返回值

IP地址(IPv4格式)，获取失败时返回null。

getGateway

```
public String getGateway()
```

功能

获取网关。

返回值

网关(IPv4格式)，获取失败时返回null。

getNetmask

```
public String getNetmask()
```

功能

获取子网掩码。

返回值

子网掩码(IPv4格式)，获取失败时返回null。

getDns1

```
public String getDns1()
```

功能

获取DNS1。

返回值

DNS1(IPv4格式)，获取失败时返回null。

getDns2

```
public String getDns2()
```

功能

获取DNS2。

返回值

DNS2(IPv4格式)，获取失败时返回null。

getServerAddress

```
public String getServerAddress()
```

功能

获取DHCP服务器地址。

返回值

DHCP服务器地址(IPv4格式)，获取失败时返回null。

getLeaseDuration

```
public int getLeaseDuration()
```

功能

获取IP租赁期。

返回值

IP租赁期，单位秒，获取失败时返回 0。

包名

titan.android.resetfactory

模块

- [恢复出厂设置服务](#)

ResetFactoryService

extends titan.android.core.TitanService

since 1.0.0

概览

恢复出厂设置工具类. 不能直接实例化ResetFactoryService, 而是通过 TitanContext.getService(Class)接口, 传入参数 ResetFactoryService.class 获取恢复出厂设置服务代理对象。

接口类

interface ResetListener

恢复出厂设置结果回调。

onResult

public void onResult(int result)

功能

恢复出厂设置过程及结果回调。

参数

- **result**: 恢复出厂设置回调结果代码。

方法说明

构造方法

不支持应用直接实例化, 而是通过TitanContext.getService(ResetFactoryService.class)接口获取实例。

resetDeep

public void resetDeep(boolean clean, ResetListener listener) throws ServiceDeathException, RemoteExecuteException

功能

深度恢复出厂设置, 包括卸载全部非系统应用、清理 (sdcard、cache、系统) 数据、重新安装客户预装应用。

参数

- **clean**: 是否选择清理系统数据, true表示清理, false表示不清理。
- **listener**: 恢复出厂设置结果回调

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 系统不支持
 - 参数错误

resetShallow

public void resetShallow(ResetListener listener) throws ServiceDeathException, RemoteExecuteException

功能

浅度恢复出厂设置, 只卸载非系统的第三方应用 (不会卸载客户的预装应用)。

参数

- **listener**: 恢复出厂设置结果回调。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 系统不支持
 - 参数错误

编程示例:

```
TitanContext titan = TitanContext.getTitanContext(context);
try {
    ResetFactoryService service = titan.getService(ResetFactoryService.class);
    //深度恢复出厂设置
    service.resetDeep(true, new ResetListener(){
        @Override
        public void onResult(int result){
        }
    });
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
} catch (RemoteExecuteException e) {
    e.printStackTrace();
}
```

包名

titan.android.systemfunction

模块

- 系统功能服务

SystemFunctionService

extends titan.android.core.TitanService

since 1.0.0

概览

SystemFunctionService是一组功能模块的服务代理接口. 不能直接实例化SystemFunctionService, 而是通过TitanContext.getService(Class)接口, 传入参数 SystemFunctionService.class 获取系统功能服务代理对象. 通过getFunction(Class)可以获得包括应用管理, 按键管理, 位置, 存储挂载等功能模块.

方法说明

构造方法

不支持应用直接实例化, 而是通过TitanContext.getService(SystemFunctionService.class)接口获取实例。

getFunction

public > T getFunction(Class clazz) throws ServiceDeathException, ServiceNotFoundException

功能

获取功能模块。

参数

- **clazz**: 服务代理对象类型, 不可为null。枚举类型有:
 - ApplicationModule.class
 - KeyModule.class
 - LocationModule.class
 - MountModule.class
 - NfcModule.class
 - ProcessModule.class
 - SettingModule.class
 - StatusBarModule.class
 - TelephonyModule
 - UsbModule

返回值

功能模块实例, 实例类型即为入参clazz指定的类型。

异常

- **ServiceNotFoundException**: clazz所指模块类型不存在时抛出此异常。
- **ServiceDeathException**: 远程服务未运行时抛出

编程示例：

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
    SystemFunctionService service = titanContext.getService(SystemFunctionService.class);
    ApplicationModule application = service.getFunction(ApplicationModule.class);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
}
```

包名

titan.android.systemfunction.application

模块

- [应用管理模块](#)
- [应用签名信息](#)
- [应用大小信息](#)
- [应用使用信息统计](#)

ApplicationModule

extends titan.android.core.TitanModule

since 1.0.0

概览

应用模块管理类. 通过SystemFunctionService.getFunction(ApplicationModule.class)获取实例。

常量

intervalType : int

public List queryUsageStats(int intervalType, long beginTime, long endTime) throws ServiceDeathException, RemoteExecuteException

功能

统计数据的时间间隔。

名称	取值	说明
USAGE_STATS_INTERVAL_DAILY	0	以天为时间间隔
USAGE_STATS_INTERVAL_WEEKLY	1	以周为时间间隔
USAGE_STATS_INTERVAL_MONTHLY	2	以月为时间间隔
USAGE_STATS_INTERVAL_YEARLY	3	以年为时间间隔
USAGE_STATS_INTERVAL_BEST	4	在给定时间范围内使用最佳的时间间隔

flags : int

public void silentlyUninstall(@NonNull String packageName, @NonNull SilentlyUninstallListener listener, int flags) throws ServiceDeathException, RemoteExecuteException

功能

卸载应用的可选值。

名称	取值	说明
DELETE_KEEP_DATA	1	卸载应用的flags取值, 表示只卸载应用, 应用数据仍保存
DELETE_ALL_USERS	2	卸载应用的flags取值, 表示为所有用户卸载应用

接口类

interface SilentlyInstallListener

静默安装回调接口。

| onInstallFinished

```
public void onInstallFinished(String packageName)
```

功能

安装成功时回调。

参数

- packageName: 应用包名。

| onInstallError

```
public void onInstallError(int error, String packageName)
```

功能

安装失败时回调。

参数

- error: 失败的错误码。
- packageName: 应用包名，可能为null。

interface SilentlyUninstallListener

静默卸载回调接口。

| onUnInstallFinished

```
public void onUnInstallFinished(String packageName)
```

功能

卸载成功时回调。

参数

- packageName: 应用包名。

| onUnInstallError

```
public void onUnInstallError(int error, String packageName)
```

功能

卸载失败时回调。

参数

- **error**: 失败的错误码。
- **packageName**: 应用包名。

interface AppSizeInfoObserver

获取应用大小信息回调接口。

onCompleted

```
public void onCompleted(AppSizeInfo appSizeInfo, boolean succeeded)
```

功能

计算应用大小结束时回调。

参数

- **appSizeInfo**: 应用大小信息。
- **succeeded**: 获取结果, **true**表示获取成功, **false**表示获取失败。

方法说明

构造方法

不支持应用直接实例化，而是通过SystemFunctionService.getFunction(ApplicationModule.class)接口获取实例。

getAppSignInfo

```
public AppSignInfo getAppSignInfo(@NonNull String packageName) throws ServiceDeathException, RemoteExecuteException
```

功能

获取应用签名信息。

参数

- **packageName**: 应用包名, 不能为**null**。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 参数错误, 如包名为**null**
 - 包名指定的应用的未安装
 - 找不到应用的apk文件
 - 系统不支持
 - 应用未签名

getAppSizeInfo

```
public void getAppSizeInfo(@NonNull String packageName, @NonNull AppSizeInfoObserver observer) throws ServiceDeathException, RemoteExecuteException
```

功能

获取应用大小.获取应用大小数据需要持续一段时间, 通过`observer` 回调返回应用大小。该接口非阻塞。

参数

- `packageName`: 应用包名, 不能为`null`。
- `observer`: 计算应用大小结束时回调, 不能为`null`。

异常

- `ServiceDeathException`: 远程服务未运行时抛出。
- `RemoteExecuteException`: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 参数错误, 如包名或回调对象为`null`
 - 包名指定的应用的未安装
 - 系统不支持

queryUsageStats

```
public List queryUsageStats(int intervalType, long beginTime, long endTime) throws ServiceDeathException, RemoteExecuteException
```

功能

获取给定时间范围的应用程序使用统计信息。

参数

- `intervalType`: 统计数据的时间间隔, 参数值`intervalType`。
- `beginTime`: 统计范围起始时间, `beginTime`在统计区间内。
- `endTime`: 统计范围结束时间, `endTime`在统计区间外。

返回值

统计信息列表, 当统计范围无使用数据时, 列表长度为0, 但不为空。

异常

- `ServiceDeathException`: 远程服务未运行时抛出。
- `RemoteExecuteException`: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 参数错误
 - 系统不支持

silentlyInstall

```
public void silentlyInstall(@NonNull String path, @NonNull SilentlyInstallListener listener) throws ServiceDeathException, RemoteExecuteException
```

功能

安装apk。安装过程会持续一段时间, 结果通过listener 回调通知。 当应用已经被安装, 如果apk是合法的, 已安装应用将被替换。该接口非阻塞。

参数

- path: apk绝对路径, 不能为null。
- listener: 安装完成通过该监听器回调, 不能为null。

异常

- ServiceDeathException: 远程服务未运行时抛出。
- RemoteExecuteException: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 参数错误
 - 系统不支持

silentlyInstall

public void silentlyInstall(@NonNull String path, @NonNull SilentlyInstallListener listener, String installerPackageName) throws ServiceDeathException, RemoteExecuteException

功能

安装apk。安装过程会持续一段时间, 结果通过listener 回调通知。 当应用已经被安装, 如果apk是合法的, 已安装应用将被替换。该接口非阻塞。

参数

- path: apk绝对路径, 不能为null。
- listener: 安装完成通过该监听器回调, 不能为null。
- installerPackageName: 正在执行安装的应用程序的可选包名称. 这标识了包来自哪个市场, 未指定时为null。

异常

- ServiceDeathException: 远程服务未运行时抛出。
- RemoteExecuteException: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 参数错误
 - 系统不支持

silentlyUninstall

public void silentlyUninstall(@NonNull String packageName, @NonNull SilentlyUninstallListener listener) throws ServiceDeathException, RemoteExecuteException

功能

卸载应用, 包括清空应用数据. 卸载应用会持续一段时间, 结果通过listener 回调通知。该接口非阻塞。

参数

- packageName: 被卸载应用的包名, 不能为null。
- listener: 卸载完成通过该监听器回调, 不能为null。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 参数错误
 - 包名指定的应用未安装
 - 系统不支持

silentlyUninstall

`public void silentlyUninstall(@NonNull String packageName, @NonNull SilentlyUninstallListener listener, int flags) throws ServiceDeathException, RemoteExecuteException`

功能

卸载应用. 卸载应用会持续一段时间, 结果通过`listener` 回调通知. 该接口非阻塞。

参数

- **packageName**: 被卸载应用的包名, 不能为`null`。
- **listener**: 卸载完成通过该监听器回调, 不能为`null`。
- **flags**: 卸载应用的可选值, 参考值`flags`。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 参数错误
 - 包名指定的应用未安装
 - 系统不支持

编程示例:

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
    SystemFunctionService service = titanContext.getService(SystemFunctionService.class);
    ApplicationModule application = service.getFunction(ApplicationModule.class);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
}
```

AppSignInfo

implements `android.os.Parcelable`

since **1.0.0**

概览

应用签名信息。

公开属性

属性名	类型	说明
rootCAId	String	根证书ID
rootCAOwner	String	根证书所有者
CAId	String	工作证书ID
signer	int	签名者
extendPermissions	List<String>	扩展权限列表

AppSizeInfo

implements `android.os.Parcelable`

since **1.0.0**

概览

应用大小信息。

公开属性

属性名	类型	说明
<code>packageName</code>	<code>String</code>	此统计数据的应用包名
<code>userHandle</code>	<code>int</code>	N/A
<code>codeSize</code>	<code>long</code>	代码大小,如APK
<code>dataSize</code>	<code>long</code>	应用程序的内部数据大小。(例如, <code>/data/data/<app>/</code>)
<code>cacheSize</code>	<code>long</code>	应用程序缓存数据大小。(例如, <code>/data/data/<app>/cache</code>)
<code>externalCodeSize</code>	<code>long</code>	保存应用程序代码的外部存储上的安全容器的大小。
<code>externalDataSize</code>	<code>long</code>	应用程序使用的外部数据的大小。(例如, <code><sdcard>/Android/data/<app></code>)
<code>externalCacheSize</code>	<code>long</code>	应用程序使用的外部高速缓存的大小(即SD卡上)。如果这是数据目录的子目录,则将从外部数据大小中减去此大小。
<code>externalMediaSize</code>	<code>long</code>	应用程序使用的外部介质大小。
<code>externalObbSize</code>	<code>long</code>	放置在外部介质的OBB的大小。

UsageStatsInfo

implements `android.os.Parcelable`

since 1.0.0

概览

指定时间范围内, 应用使用信息统计。

公开属性

属性名	类型	说明
<code>packageName</code>	<code>String</code>	包含使用情况统计信息的应用包名称
<code>beginTimeStamp</code>	<code>long</code>	统计信息的开始时间, 以毫秒为单位
<code>endTimeStamp</code>	<code>long</code>	统计信息的结束时间, 以毫秒为单位
<code>lastTimeUsed</code>	<code>long</code>	最后一次使用此应用的时间点, 以毫秒为单位
<code>totalTimeInForeground</code>	<code>long</code>	应用在系统前台的总时间, 以毫秒为单位

包名

titan.android.systemfunction.key

模块

- [按键管理模块](#)

包名

titan.android.systemfunction.location

模块

- [位置管理模块](#)

LocationModule

extends titan.android.core.TitanModule

since 1.0.0

概览

位置模块管理类。通过SystemFunctionService.getFunction(LocationModule.class)获取实例。

常量

mode : int

boolean titan.android.systemfunction.location.LocationModule.setLocationMode(int mode) throws ServiceDeathException

功能

位置模式的可选值。

名称	取值	说明
LOCATION_MODE_OFF	0	关闭位置服务
LOCATION_MODE_SENSORS_ONLY	1	仅限设备, 使用GPS确定位置
LOCATION_MODE_BATTERY_SAVING	2	耗电量低, 使用WLAN和移动网络确定位置
LOCATION_MODE_HIGH_ACCURACY	3	准确度高, 使用GPS、WLAN和移动网络确定位置

方法说明

构造方法

不支持应用直接实例化，而是通过SystemFunctionService.getFunction(LocationModule.class)接口获取实例。

getLocationMode

public int getLocationMode() throws ServiceDeathException, RemoteExecuteException

功能

获取当前位置模式。

返回值

当前位置模式。参考值[mode](#)

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

setLocationMode

`public boolean setLocationMode(int mode) throws ServiceDeathException`

功能

设置位置模式。错误码可通过{@link TitanContext#getLastError()}查询。

参数

- **mode**: 位置模式。参考值[mode](#)

返回值

设置成功, 返回 **true**, 否则返回**false**。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 系统不支持

编程示例:

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
    SystemFunctionService service = titanContext.getService(SystemFunctionService.class);
    LocationModule location = service.getFunction(LocationModule.class);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
}
```

包名

titan.android.systemfunction.mount

模块

- [挂载管理模块](#)

MountModule

extends titan.android.core.TitanModule

since 1.0.0

概览

外置存储设备挂载模块管理类. 通过SystemFunctionService.getFunction(MountModule.class)获取实例。

接口类

interface StorageEventListener

存储器事件监听器, 用于监听存储器挂载/解除挂载事件。

onUsbMassStorageConnectionChanged

```
public void onUsbMassStorageConnectionChanged(boolean connected)
```

功能

当检测到USB大容量存储器的状态发生变化时回调。

参数

- connected: USB大容量存储器已经连接上时为true, 否则为false。

onStorageStateChanged

```
public void onStorageStateChanged(String path, String oldState, String newState)
```

功能

存储器状态改变时回调。

参数

- path: 存储器文件系统路径。
- oldState: 存储器之前的状态, 状态码与android.os.Environment.getExternalStorageState()一致。
- newState: 存储器当前的状态, 状态码与android.os.Environment.getExternalStorageState()一致。

方法说明

构造方法

不支持应用直接实例化, 而是通过SystemFunctionService.getFunction(MountModule.class)接口获取实例。

mountSDcard

`public void mountSDcard() throws ServiceDeathException, RemoteExecuteException`

功能

挂载sdcard存储器. 挂载和解除挂载存储器是一个异步过程, 如果应用需要获得存储器状态的反馈, 可以通过注册监听器 `registerStorageEventListener(StorageEventListener)`进行结果监听。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 系统不支持

mountUdisk

`public void mountUdisk() throws ServiceDeathException, RemoteExecuteException`

功能

挂载U盘. 挂载和解除挂载存储器是一个异步过程, 如果应用需要获得存储器状态的反馈, 可以通过注册监听器 `registerStorageEventListener(StorageEventListener)`进行结果监听。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 系统不支持

unmountSDcard

`public void unmountSDcard() throws ServiceDeathException, RemoteExecuteException`

功能

解除挂载sdcard存储器. 挂载和解除挂载存储器是一个异步过程, 如果应用需要获得存储器状态的反馈, 可以通过注册监听器 `registerStorageEventListener(StorageEventListener)`进行结果监听。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 系统不支持

unmountUdisk

`public void unmountUdisk() throws ServiceDeathException, RemoteExecuteException`

功能

解除挂载U盘. 挂载和解除挂载存储器是一个异步过程, 如果应用需要获得存储器状态的反馈, 可以通过注册监听器 `registerStorageEventListener(StorageEventListener)` 进行结果监听。

异常

- `ServiceDeathException`: 远程服务未运行时抛出。
- `RemoteExecuteException`: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 系统不支持

registerStorageEventListener

```
public boolean registerStorageEventListener(StorageEventListener listener) throws ServiceDeathException
```

功能

注册一个全局的存储器事件监听器, 用于接收存储器挂载状态改变时的异步通知. 由于挂载是一个异步过程, 应用应该在调用挂载 `mountSDcard()`, `mountUdisk()` 或解除挂载 `unmountSDcard()`, `unmountUdisk()` 方法调用前注册监听器, 在挂载状态改变时接收事件通知. 当应用不再需要监听存储器事件时需要调用 `unregisterStorageEventListener()` 方法取消监听, 否则如果有其他线程或其他应用操作存储器时, 应用仍能接收到事件通知。

参数

- `listener`: 存储器事件监听器。

返回值

注册成功返回 `true`, 否则返回 `false`。

异常

- `ServiceDeathException`: 远程服务未运行时抛出。

unregisterStorageEventListener

```
public boolean unregisterStorageEventListener() throws ServiceDeathException
```

功能

反注册一个全局的存储器事件监听器, 当应用不再需要监听存储器事件时需要取消监听, 否则如果有其他线程或其他应用操作存储器时, 应用仍能接收到事件通知。

返回值

反注册成功返回 `true`, 否则返回 `false`。

异常

- `ServiceDeathException`: 远程服务未运行时抛出。

编程示例:

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
```



```
        SystemFunctionService service = titanContext.getService(SystemFunctionService.class);
        MountModule mount = service.getFunction(MountModule.class);
    } catch (ServiceNotFoundException e) {
        e.printStackTrace();
    } catch (ServiceDeathException e) {
        e.printStackTrace();
    }
}
```

包名

titan.android.systemfunction.nfc

模块

- [NFC管理模块](#)

NfcModule

extends titan.android.core.TitanModule

since 1.0.0

概览

NFC模块管理类。通过SystemFunctionService.getFunction(NfcModule.class)获取实例。

常量

state : int

public int getNfcState() throws ServiceDeathException, RemoteExecuteException

功能

NFC模块状态可选值。

名称	取值	说明
NFC_NOT_SUPPORTED	0	设备不支持NFC
NFC_ENABLED	1	启用NFC功能
NFC_DISENABLED	2	关闭NFC功能

方法说明

构造方法

不支持应用直接实例化，而是通过SystemFunctionService.getFunction(NfcModule.class)接口获取实例。

getNfcState

public int getNfcState() throws ServiceDeathException, RemoteExecuteException

功能

获取NFC状态。

返回值

当前位置模式。参考值[state](#)

异常

- ServiceDeathException: 远程服务未运行时抛出。

- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

setNfcState

`public boolean setNfcState(int state) throws ServiceDeathException`

功能

设置NFC状态。错误码可通过{@link TitanContext#getLastError()}查询。

参数

- **state**: 位置模式。取值有:
 - NFC_ENABLED
 - NFC_DISENABLED

返回值

设置成功, 返回 **true**, 否则返回**false**。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 参数错误
 - 设备不支持
 - 系统不支持

编程示例:

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
    SystemFunctionService service = titanContext.getService(SystemFunctionService.class);
    NfcModule nfc = service.getFunction(NfcModule.class);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
}
```

包名

titan.android.systemfunction.process

模块

- [进程管理模块](#)

ProcessModule

extends titan.android.core.TitanModule

since 1.0.0

概览

应用进程模块管理类。通过SystemFunctionService.getFunction(ProcessModule.class)获取实例。

方法说明

构造方法

不支持应用直接实例化，而是通过SystemFunctionService.getFunction(ProcessModule.class)接口获取实例。

clearRecentTasks

public boolean clearRecentTasks() throws ServiceDeathException

功能

清空杀死同一用户最近启动的后台任务. 远端服务限制一次清空任务最大个数为100, 如果后台任务个数大于100, 需要再次执行此方法, 直到清空干净. 错误码可通过TitanContext.getLastError()查询。

返回值

成功清空任务返回true, 否则返回false

异常

- ServiceDeathException: 远程服务未运行时抛出。

编程示例：

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
    SystemFunctionService service = titanContext.getService(SystemFunctionService.class);
    ProcessModule process = service.getFunction(ProcessModule.class);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
}
```


包名

titan.android.systemfunction.setting

模块

- [设置管理模块](#)
- [输入法信息](#)
- [语言信息](#)
- [时区信息](#)

SettingModule

extends titan.android.core.TitanModule

since 1.0.0

概览

系统设置相关的管理类.通过此接口,可以设置系统语言,时间/时区,输入法等,以及获取系统当前有关设置信息。通过SystemFunctionService.getFunction(SettingModule.class)获取实例。

方法说明

构造方法

不支持应用直接实例化,而是通过SystemFunctionService.getFunction(SettingModule.class)接口获取实例。

getCurrentIME

```
public IMEInfo getCurrentIME() throws ServiceDeathException, RemoteExecuteException
```

功能

获取系统当前启用的输入法信息。

返回值

输入法元信息。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常,以下情况会抛出异常
 - 系统不支持
 - 未设置输入法

setCurrentIME

```
public boolean setCurrentIME(@NonNull String packageName, @NonNull String className) throws ServiceDeathException
```

功能

设置输入法。

参数

- **packageName**: 输入法包名,不能为**null**。
- **className**: 输入法全路径包名,不能为**null**。

返回值

设置成功返回true, 否则返回false, 返回false时可通过 TitanContext.getLastError()获取错误码。

异常

- ServiceDeathException: 远程服务未运行时抛出。

getInstalledIMEList

public List getInstalledIMEList() throws ServiceDeathException, RemoteExecuteException

功能

获取已安装的输入法列表。

返回值

输入法列表, 当未安装输入法时, 列表长度为0, 但不为空。

异常

- ServiceDeathException: 远程服务未运行时抛出。
- RemoteExecuteException: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

setSystemLanguage

public boolean setSystemLanguage(@NonNull String languageCode, @NonNull String countryCode) throws ServiceDeathException

功能

设置系统语言。

参数

- languageCode: 语言代码, 不能为null。
- countryCode: 国家或地区代码, 不能为null。

返回值

设置成功返回true, 否则返回false, 返回false时可通过 TitanContext.getLastError()获取错误码。

异常

- ServiceDeathException: 远程服务未运行时抛出。

getLanguageList

public List getLanguageList() throws ServiceDeathException, RemoteExecuteException

功能

获取系统支持的语言列表。

返回值

语言列表。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

| **getCurrentScreenTimeout**

`public int getCurrentScreenTimeout() throws ServiceDeathException, RemoteExecuteException`

功能

获取屏幕超时休眠时间。

返回值

休眠时间, 单位毫秒。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

| **setScreenTimeout**

`public boolean setScreenTimeout(int timeout) throws ServiceDeathException`

功能

设置屏幕超时休眠时间, 当屏幕一段时间无操作时, 系统进入休眠。

参数

- **timeout**: 超时时间, 表示一段时间后无操作, 系统进入休眠, 单位毫秒。参考值:
 - -1:永不休眠
 - 15000:15秒后休眠
 - 30000:30秒后休眠
 - 60000:1分钟后休眠
 - 120000:2分钟后休眠
 - 300000:5分钟后休眠
 - 600000:10分钟后休眠
 - 1800000:30分钟后休眠

返回值

设置成功返回true, 否则返回false, 返回false时可通过 `TitanContext.getLastError()` 获取错误码。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。

getScreenTimeoutList

`public List getScreenTimeoutList() throws ServiceDeathException, RemoteExecuteException`

功能

获取系统支持的屏幕超时休眠时间列表。

返回值

休眠时间列表。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

getCurrentTimeZone

`public TimeZoneInfo getCurrentTimeZone() throws ServiceDeathException, RemoteExecuteException`

功能

获取系统当前时区。

返回值

时区元信息。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

setTimeZone

`public boolean setTimeZone(@NonNull String id) throws ServiceDeathException`

功能

设置系统时区。

参数

- **id**: 时区id, 不可为null

返回值

设置成功返回true, 否则返回false, 返回false时可通过 `TitanContext.getLastError()` 获取错误码。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。

setAutomaticTimeZoneEnabled

public boolean setAutomaticTimeZoneEnabled(boolean enabled) throws ServiceDeathException, RemoteExecuteException

功能

启用/禁用自动确定时区。

参数

- **enabled:** true表示启用, false表示禁用

返回值

设置成功返回true, 否则返回false, 返回false时可通过 TitanContext.getLastError()获取错误码。

异常

- **ServiceDeathException:** 远程服务未运行时抛出。

getTimeZoneList

public List getTimeZoneList() throws ServiceDeathException, RemoteExecuteException

功能

获取系统支持的时区列表。

返回值

时区列表。

异常

- **ServiceDeathException:** 远程服务未运行时抛出。
- **RemoteExecuteException:** 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

getSystemTime

public long getSystemTime() throws ServiceDeathException, RemoteExecuteException

功能

获取系统时间. 无论系统的时区如何, 此方法始终返回UTC时间。这通常被称为“Unix时间”或“纪元时间”. 使用 java.text.DateFormat实例格式化此时间。

返回值

返回自1970年1月1日00: 00: 00.0 UTC以来的当前时间（以毫秒为单位）。

异常

- **ServiceDeathException:** 远程服务未运行时抛出。
- **RemoteExecuteException:** 接口执行过程异常, 以下情况会抛出异常

- 系统不支持

| **setSystemTime**

`public boolean setSystemTime(long time) throws ServiceDeathException`

功能

设置系统时间。

参数

time : 系统时间, 以毫秒为单位。

返回值

设置成功返回**true**, 否则返回**false**, 返回**false**时可通过 `TitanContext.getLastError()`获取错误码。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。

| **setAutomaticTimeEnabled**

`public boolean setAutomaticTimeEnabled(boolean enabled) throws ServiceDeathException, RemoteExecuteException`

功能

启用/禁用自动确定时间。

参数

enabled: **true**表示启用, **false**表示禁用。

返回值

设置成功返回**true**, 否则返回**false**, 返回**false**时可通过 `TitanContext.getLastError()`获取错误码。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。

| **setCustomBootAnimation**

`public boolean setCustomBootAnimation(@NonNull String zipPath) throws ServiceDeathException`

功能

自定义开机动画. 设置开机动画接口不检查**zip**包内的动画文件是否合法, 需要开发者自行检查是否符合动画规则。
可通过**resetBootAnimation()** 恢复默认值。

参数

zipPath: 动画文件绝对路径, 不可为**null**。

返回值

设置成功返回true, 否则返回false, 返回false时可通过 TitanContext.getLastError()获取错误码。

异常

- ServiceDeathException: 远程服务未运行时抛出。

resetBootAnimation

public boolean resetBootAnimation() throws ServiceDeathException

功能

恢复开机动画, 使用系统默认的开机动画。

返回值

恢复成功返回true, 否则返回false, 返回false时可通过 TitanContext.getLastError()获取错误码。

异常

- ServiceDeathException: 远程服务未运行时抛出。

setCellularDataEnabled

public boolean setCellularDataEnabled(boolean enabled) throws ServiceDeathException

功能

启用/禁用移动网络。

参数

enabled : true表示启用, false表示禁用

返回值

设置成功返回true, 否则返回false, 返回false时可通过 TitanContext.getLastError()获取错误码。

异常

- ServiceDeathException: 远程服务未运行时抛出。

isCellularDataEnabled

public boolean isCellularDataEnabled() throws ServiceDeathException, RemoteExecuteException

功能

检查移动网络是否启用。

返回值

启用返回true, 禁用返回false。

异常

- ServiceDeathException: 远程服务未运行时抛出。

- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

编程示例:

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
    SystemFunctionService service = titanContext.getService(SystemFunctionService.class);
    SettingModule setting = service.getFunction(SettingModule.class);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
}
```


IMEInfo

implements `android.os.Parcelable`

since 1.0.0

概览

该类用于指定输入法的元信息。

公开属性

属性名	类型	说明
id	String	识别输入方法的唯一字符串Id, 由输入法组件生成的, 如: <code>com.android.inputmethod.latin.LatinIME</code>
packageName	String	输入法包名, 如: <code>com.android.inputmethod.latin</code>
serviceName	String	输入法服务组件名称, 如: <code>com.android.inputmethod.latin.LatinIME</code>
className	String	输入法服务组件全路径类名, 如: <code>com.android.inputmethod.latin.LatinIME</code>
settingsActivity	String	返回为输入方法提供设置UI界面的类名称, 如: <code>com.android.inputmethod.latin.settings.SettingsActivity</code> , 如果没有与输入方法关联的设置活动, 则为null
shortClassName	String	输入法服务组件类名, 如: <code>LatinIME</code>
permission	String	能够访问输入法服务所需权限的可选名称. 来自"permission" 属性. 如: <code>android.permission.BIND_INPUT_METHOD</code>
subtypeCount	int	输入法支持的语言数目
isEnabled	boolean	输入法是否启用
label	String	输入法label
subtypeDisplayName	String	输入法当前使用的语言名称

LanguageInfo

implements `android.os.Parcelable`

since 1.0.0

概览

该类用于指定语言的元信息。

公开属性

属性名	类型	说明
<code>languageCode</code>	<code>String</code>	语言代码：如:zh
<code>countryCode</code>	<code>String</code>	国家或地区代码，如:CN
<code>label</code>	<code>String</code>	语言名称：如:中文（简体）
<code>info</code>	<code>int</code>	语言信息：如:zh_CN-中文（简体）

TimeZoneInfo

implements `android.os.Parcelable`

since **1.0.0**

概览

该类用于指定时区的元信息。

公开属性

属性名	类型	说明
id	String	时区代码, 如:America/Los_Angeles, GMT-08:00 or UTC
displayName	String	时区名称
label	String	语言名称: 如:中文(简体)
DSTSavings	int	此时区的最新夏令时(以毫秒为单位), 相对于此时区的常规UTC偏移量.对于使用夏令时的时区, 返回3600000(1小时), 对于不使用夏令时的时区, 例如Australia / Lord_Howe, 使用其他值.)
rawOffset	int	此时区标准时间与UTC的偏移量(以毫秒为单位)

包名

titan.android.systemfunction.statusbar

模块

- [状态栏管理模块](#)

StatusBarModule

extends titan.android.core.TitanModule

since 1.0.0

概览

状态栏管理模块.对物理按键如电源键及虚拟按键的管理接口. 通过 `SystemFunctionService.getFunction(StatusBarModule.class)` 获取实例。

方法说明

构造方法

不支持应用直接实例化，而是通过 `SystemFunctionService.getFunction(StatusBarModule.class)` 接口获取实例。

isPanelExpandEnabled

`public boolean isPanelExpandEnabled() throws ServiceDeathException, RemoteExecuteException`

功能

检查状态栏下拉是否启用。

返回值

状态栏下拉启用返回 `true`, 否则返回 `false`

异常

- `ServiceDeathException`: 远程服务未运行时抛出。
- `RemoteExecuteException`: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

setPanelExpandEnabled

`public boolean setPanelExpandEnabled(boolean enabled) throws ServiceDeathException`

功能

设置状态栏下拉是否启用。

参数

- `enabled`: 启用下拉为 `true`, 禁用下拉为 `false`。

返回值

设置成功返回 `true`, 否则返回 `false`. 返回 `false` 时可通过 `TitanContext.getLastError()` 获取错误码。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。

编程示例:

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
    SystemFunctionService service = titanContext.getService(SystemFunctionService.class);
    StatusBarModule statusBar = service.getFunction(StatusBarModule.class);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
}
```

包名

titan.android.systemfunction.telephony

模块

- [Telephony](#)管理模块

TelephonyModule

extends titan.android.core.TitanModule

since 1.0.0

概览

Telephony服务管理模块.对物理按键如电源键及虚拟按键的管理接口. 通过 `SystemFunctionService.getFunction(TelephonyModule.class)` 获取实例。

常量

networkType : int

public boolean setPreferredNetworkType(int networkType) throws ServiceDeathException

功能

网络首选项可选值。

名称	取值	说明
NETWORK_MODE_WCDMA_PREF	0	GSM/WCDMA (首选 WCDMA)
NETWORK_MODE_GSM_ONLY	1	仅GSM
NETWORK_MODE_WCDMA_ONLY	2	仅WCDMA
NETWORK_MODE_GSM_UMTS	3	GSM/WCDMA (根据 PRL, 自动模式)
NETWORK_MODE_CDMA	4	CDMA and EvDo (根据 PRL, 自动模式)
NETWORK_MODE_CDMA_NO_EVDO	5	仅CDMA
NETWORK_MODE_EVDO_NO_CDMA	6	仅EvDo
NETWORK_MODE_GLOBAL	7	GSM/WCDMA, CDMA, EvDo (根据PRL, 自动模式)
NETWORK_MODE_LTE_CDMA_EVDO	8	LTE, CDMA, EvDo
NETWORK_MODE_LTE_GSM_WCDMA	9	LTE, GSM/WCDMA
NETWORK_MODE_LTE_CDMA_EVDO_GSM_WCDMA	10	LTE, CDMA, EvDo, GSM/WCDMA
NETWORK_MODE_LTE_ONLY	11	LTE 唯一模式
NETWORK_MODE_LTE_WCDMA	12	LTE/WCDMA

NETWORK_MODE_TD_SCDMA_ONLY	13	仅TD-SCDMA
NETWORK_MODE_TD_SCDMA_WCDMA	14	TD-SCDMA, WCDMA
NETWORK_MODE_TD_SCDMA_LTE	15	TD-SCDMA, LTE
NETWORK_MODE_TD_SCDMA_GSM	16	TD-SCDMA, GSM
NETWORK_MODE_TD_SCDMA_GSM_LTE	17	TD-SCDMA,GSM, LTE
NETWORK_MODE_TD_SCDMA_GSM_WCDMA	18	TD-SCDMA, GSM/WCDMA
NETWORK_MODE_TD_SCDMA_WCDMA_LTE	19	TD-SCDMA, WCDMA, LTE
NETWORK_MODE_TD_SCDMA_GSM_WCDMA_LTE	20	TD-SCDMA, GSM/WCDMA, LTE
NETWORK_MODE_TD_SCDMA_CDMA_EVDO_GSM_WCDMA	21	TD- SCDMA,EvDo,CDMA, GSM/WCDMA
NETWORK_MODE_TD_SCDMA_LTE_CDMA_EVDO_GSM_WCDMA	22	TD-SCDMA/ LTE/GSM/ WCDMA, CDMA, EvDo
NETWORK_MODE_LTE_CDMA_EVDO_GSM	23	LTE/CDMA/EvDo/GSM

方法说明

构造方法

不支持应用直接实例化，而是通过SystemFunctionService.getFunction(TelephonyModule.class)接口获取实例。

getSimRecord5F01

public String getSimRecord5F01() throws ServiceDeathException, RemoteExecuteException

功能

获取SIM卡特定位置5F01位置上记录信息.(通联特定SIM卡)。

返回值

5F01位置记录信息。

异常

- ServiceDeathException: 远程服务未运行时抛出。
- RemoteExecuteException: 接口执行过程异常, 以下情况会抛出异常
 - SIM卡状态错误
 - 系统不支持

getSimRecord5F2

public String getSimRecord5F02() throws ServiceDeathException, RemoteExecuteException

功能

获取SIM卡特定位置5F02位置上记录信息.(通联特定SIM卡)。

返回值

5F02位置记录信息。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - SIM卡状态错误
 - 系统不支持

getIMEI

```
public String getIMEI(int slot) throws ServiceDeathException, RemoteExecuteException
```

功能

获取指定卡槽的IMEI。

参数

- **slot**: 卡槽id, 取值范围[0, 1]

返回值

设备IMEI值, 卡槽不被支持, 返回null。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

getDeviceId

```
public String getDeviceId(int slot) throws ServiceDeathException, RemoteExecuteException
```

功能

获取设备ID, 例如, GSM的IMEI和CDMA电话的MEID或ESN。

参数

- **slot**: 卡槽id, 取值范围[0, 1]

返回值

设备ID值, 卡槽不被支持, 返回null。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常

- 系统不支持

getPreferredNetworkType

public int getPreferredNetworkType() throws ServiceDeathException, RemoteExecuteException

功能

获取网络首选项类型。

返回值

网络首选项类型。参考值[networkType](#)。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

setPreferredNetworkType

public boolean setPreferredNetworkType(int networkType) throws ServiceDeathException

功能

设置网络首选项类型。

参数

- **networkType**: 网络首选项类型。参考值[networkType](#)。

返回值

设置成功返回true, 否则返回false, 返回false时可通过 `TitanContext.getLastError()`获取错误码。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。

编程示例:

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
    SystemFunctionService service = titanContext.getService(SystemFunctionService.class);
    TelephonyModule telephony = service.getFunction(TelephonyModule.class);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
}
```


包名

titan.android.systemfunction.usb

模块

- [USB管理模块](#)

UsbModule

extends titan.android.core.TitanModule

since 1.0.0

概览

USB模块管理类。通过SystemFunctionService.getFunction(UsbModule.class)获取实例。

枚举类

UsbMode

USB功能枚举

- NONE：不设置功能
- MTP：MTP(默认值)
- PTP：PTP
- MASS_STORAGE：MASS_STORAGE

方法说明

构造方法

不支持应用直接实例化，而是通过SystemFunctionService.getFunction(UsbModule.class)接口获取实例。

isFunctionEnabled

public boolean isFunctionEnabled(@NonNull UsbMode usbMode) throws ServiceDeathException, RemoteExecuteException

功能

检查USB功能是否启用。

参数

- usbMode：USB功能模式, 不可为null。参考UsbMode。

返回值

功能已启用返回true, 否则返回false

异常

- ServiceDeathException：远程服务未运行时抛出。
- RemoteExecuteException：接口执行过程异常, 以下情况会抛出异常
 - 系统不支持

setCurrentFunction

public void setCurrentFunction(UsbMode usbMode) throws ServiceDeathException, RemoteExecuteException

功能

设置当前USB功能. 应用可注册广播监听设置结果。

```
public final BroadcastReceiver setUsbFunctionReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (action.equals(UsbManager.ACTION_USB_STATE)) {
            boolean connected = intent.getBooleanExtra("connected", false);
            boolean configured = intent.getBooleanExtra("configured", false);
            boolean function_adb = intent.getBooleanExtra("adb", false);
            boolean function_mtp = intent.getBooleanExtra("mtp", false);
            boolean function_ptp = intent.getBooleanExtra("ptp", false);
        }
    }
};
```

注册动态广播:

```
@Override
public void onResume() {
    super.onResume();
    IntentFilter intentFilter = new IntentFilter();
    intentFilter.addAction(UsbManager.ACTION_USB_STATE);
    getActivity().registerReceiver(setUsbFunctionReceiver, intentFilter);
}
```

参数

- **usbMode**: USB功能模式, 不可为null。参考[UsbMode](#)。

异常

- **ServiceDeathException**: 远程服务未运行时抛出。
- **RemoteExecuteException**: 接口执行过程异常, 以下情况会抛出异常
 - 应用无权限
 - 参数错误
 - 系统不支持

编程示例:

```
TitanContext titanContext = TitanContext.getTitanContext(context);
try {
    SystemFunctionService service = titanContext.getService(SystemFunctionService.class);
    UsbModule usb = service.getFunction(UsbModule.class);
} catch (ServiceNotFoundException e) {
    e.printStackTrace();
} catch (ServiceDeathException e) {
    e.printStackTrace();
}
```

