

保密信息

梅特勒-托利多(常州)测量技术有限公司
Mettler-Toledo (Changzhou) Measurement Technology Ltd.

地址: 江苏省常州市新北区
太湖西路111号
电话: 86-0519-86642040
传真: 86-0519-86641991
邮编: 213125

Address: 111 Taihu West Road,
Changzhou, Jiangsu
Tel: 86-0519-86642040
Fax: 86-0519-86641991
Zip: 213125

www.mt.com

MT Android 软件开发指南



保密信息

梅特勒-托利多(常州)测量技术有限公司
Mettler-Toledo (Changzhou) Measurement Technology Ltd.

地址: 江苏省常州市新北区
太湖西路111号
电话: 86-0519-86642040
传真: 86-0519-86641991
邮编: 213125

Address: 111 Taihu West Road,
Changzhou, Jiangsu
Tel: 86-0519-86642040
Fax: 86-0519-86641991
Zip: 213125

www.mt.com

修订历史

日期	版本	作者	备注
2019-04-25	1.0	杨宇	创建
2020-04-09	1.1	杨宇	1. 新增 2.7 ADB 调试 2. 修改 4.1 多打印机支持说明 ; 3. 新增 4.4.2 热插拔事件通知 ; 4. 删除 4.10.6 设置打印模式 5. 增加 错误码-14 和-1004
2020-04-23	1.2	杨宇	1. 增加 开始打印 接口 2. 增加 错误码-1005
2020-05-08	1.3	杨宇	1. 增加 系统控制接口
2020-05-29	1.4	杨宇	1. 增加 设置字体缩放系数

保密信息

梅特勒-托利多(常州)测量技术有限公司
Mettler-Toledo (Changzhou) Measurement Technology Ltd.

地址: 江苏省常州市新北区
太湖西路111号
电话: 86-0519-86642040
传真: 86-0519-86641991
邮编: 213125

Address: 111 Taihu West Road,
Changzhou, Jiangsu
Tel: 86-0519-86642040
Fax: 86-0519-86641991
Zip: 213125

www.mt.com

目 录

1. 平台介绍	5
1.1 外观	5
1.2 硬件	5
1.3 软件	6
1.4 概念	6
2. 方案说明	7
2.1 组织方式	7
2.2 整体架构	7
2.3 API 设计	8
2.4 开发流程	8
2.5 文件清单	11
2.6 安装路径	11
2.7 ADB 调试	11
3. 称重接口	14
3.1 概述	14
3.2 获取称重 API 实例	14
3.3 建立连接	14
3.4 回调接口说明	15
3.5 设置称量皮重	18
3.6 设置预置皮重	19
3.7 清除皮重	20
3.8 手动清零	20
4. 打印接口	21
4.1 调用打印机说明	21

4.2	获取称重 API 实例	21
4.3	建立连接	22
4.4	回调接口说明	22
4.5	查询打印机状态	24
4.6	打印文本	25
4.7	打印多列文本	26
4.8	打印位图	26
4.9	打印空白行	27
4.10	开始打印	27
4.11	设置默认值	28
4.12	辅助功能	30
4.13	钱箱相关	34
5.	系统接口	34
5.1	沉浸式接口说明	34
5.2	修改屏幕密度	34
6.	返回码及含义	35
7.	代码实例	37

1. 平台介绍

梅特勒-多利多（下面简称 MT）作为一家历史悠久的称重设备供应商，紧跟市场发展，实时推出基于 Android 系统的多款产品，同时，MT 也积极推动同第三方 POS 厂商和经销商合作，将 MT 的硬件产品同友商优秀、成熟的收银和管理软件相结合，希望能迅速占领零售称重市场，实现双赢。

MTAndroid 系统，采用 Android 7.1 操作系统，集成 MT 专有称重和打印服务和精心设计的 API 接口，使第三方应用程序能通过本平台轻松的完成称重、打印等功能，而不需要关注称重本身的复杂逻辑，为客户提供优秀的产品。

1.1 外观



图 1. 正面效果图

1.2 硬件

MTAndroid 系统的硬件平台主要配置参数如下：

硬件	备注
CPU	ARM Rockchip RK3399, Dual-core Cortex-A72 + Quad-core Cortex-A53 Clock speed up to 1.8GHz and 1.4GHz respectively
Memory	2/4GB
Flash	EMMC 8/16/32G

保密信息

梅特勒-托利多(常州)测量技术有限公司
Mettler-Toledo (Changzhou) Measurement Technology Ltd.

地址: 江苏省常州市新北区
太湖西路111号
电话: 86-0519-86642040
传真: 86-0519-86641991
邮编: 213125

Address: 111 Taihu West Road,
Changzhou, Jiangsu
Tel: 86-0519-86642040
Fax: 86-0519-86641991
Zip: 213125

www.mt.com

以太网	10/100/1000M 自适应
Wifi	802.11a/b/g/n/ac 2.4GHz/5GHz Bluetooth 4.2 (support BLE)
主显示屏尺寸	15.6 英寸 分辨率 1920*1080
客显示屏尺寸	15.6 英寸 分辨率 1920*1080
USB2.0	3
USB3.0	1
串口	1
钱箱接口	1
Audio	1

1.3 软件

主要的软件版本信息如下:

软件	版本信息
Android	7.1.2
IDE	Android Studio3.6
JDK	jdk1.8.0_191
JRE	1.8.0
compileSdkVersion	28
minSdkVersion	25

注: 以上软件均为标准 Android Studio 开发组件, 请自行安装。

1.4 概念

皮重 Tare: 主要是指商品外包装的重量, 如: 装水果的篮子、装鱼虾的盆等, 一般不算销售产品的重量;

净重 Net: 指单纯商品的重量, 不包括皮重;

毛重 Gross: 净重加皮重之和;

预置皮重 Preset Tare: 某些商品皮重固定, 在称量商品前可以通过“输入”的方式设置皮重值;

最小称量: 小于该载荷值时, 称重结果可能产生较大的相对误差; 比如: 精度是 2g 的秤, 其最小量程可能是 40g;

最大量程: 不考虑添加皮重的最大称重能力;

App: 本文中使用 App 表示第三方开发的应用程序, 如 POS 软件;

2. 方案说明

2.1 组织方式

为了便捷性维护性和优秀的扩展性, MT 称重服务和打印服务设计以独立 Service 组件 APP 方式安装到 Android 系统中, 其它厂商 APP 通过 MT 提供的开发 API 与之做 IPC 通信, 这样明显的优点是多方应用独立升级和维护自己的 APP, 基本上互不依赖, 实现解耦。

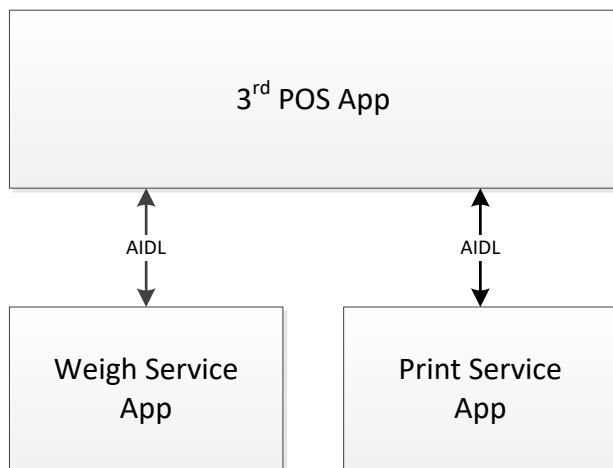


图 2. 系统结构图

2.2 整体架构

整体架构上采用了 MVP 设计模式, MT 打印和称重服务负责 Model 以及自身功能的 Presenter 工作, APP 负责 View 显示和其它业务处理; 在 Presenter 和 View 之间采用接口和回调方式完成数据的更新和传递;

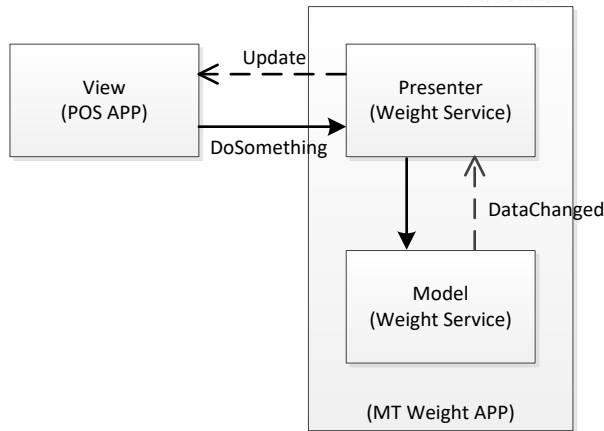
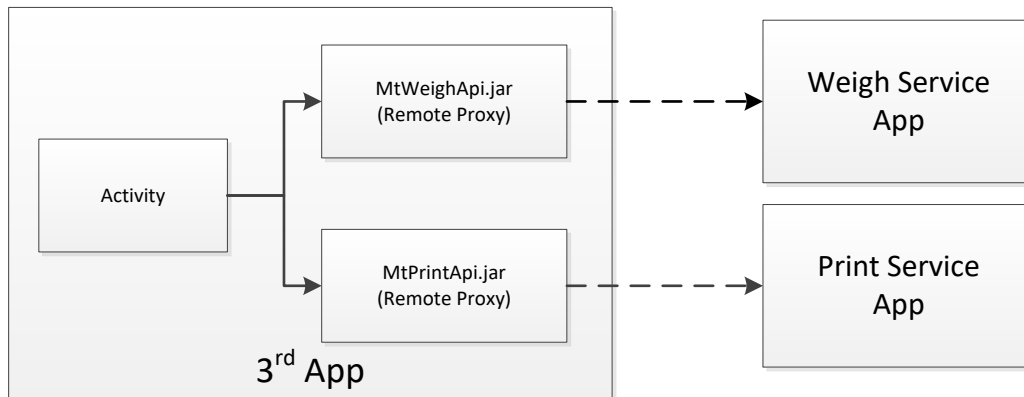


图 3. 称重服务设计架构图

2.3 API 设计

为了简化 APP 开发的难度和差异性，MT 采用远程代理的设计，对客户端需要编写的 Remote Service 相关代码进行了封装处理，合作方 APP 只需包含 MT 提供的称重和打印 Api.jar 文件，即可像调用本地函数一样使用跨进程的称重和打印服务，提升开发效率。



2.4 开发流程

称重服务和打印服务的流程基本相同，主要有如下几个步骤：

1. 获取 MtWeighApi 和 MtPrintApi 对象实例；

例如：

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}

```



```
setContentView(R.layout.activity_main);

mMtWeightService = MtWeighApi.getInstance();

mMtPrintService = MtPrintApi.getInstance();

}
```

2. 实现 View 的回调接口;

在 MVP 模式中, 当 Model 数据变化时触发 Presenter 刷新 UI, View 需要实现刷新 UI 的接口供 Presenter 调用; 对应称重和打印服务, APP 需要实现如下两个 View 接口: IMtWeighView 和 IMtPrintView, 说明详见[称重回调接口](#)和[打印回调接口](#)。

例如:

```
public class MainActivity extends AppCompatActivity implements IMtWeighView,
IMtPrintView {

    private MtWeighApi mMtWeightService;

    private MtPrintApi mMtPrintService;

    .....

    @Override

    public void onMtWeighServiceConnected() {

        initWeightView();

    }

    @Override

    public void onWeightChanged(String s) {

        runOnUiThread(() -> updateUiWeight(s));

    }

    .....

}
```

实现了 onWeightChanged 接口后, UI 上重量显示部分会自动被更新, 不再需要 Client 去主动查询重量信息。

3. 建立到称重和打印服务的连接;

由于 Android 本身机制, 连接建立过程是异步的 (非阻塞), 即发送完连接请求后函数即返回, 实际连接建立成功大概要延迟 200~400ms, 系统通过回调接口通知 Client 连接建立成功, 因此, 需要注册响应 (步骤 2) 的回调接口;

例如:

```
@Override

protected void onCreate(Bundle savedInstanceState) {

    .....

}
```

```
mMtWeightService = MtWeighApi.getInstance();
mMtPrintService = MtPrintApi.getInstance();

mMtWeightService.connectToService(this, this);
mMtPrintService.connectToService(this, this);
}
@Override
public void onMtWeighServiceConnected() {
    doSomeInit();
}
@Override
public void onMtPrintServiceConnected() {
    initPrintView();
}
@Override
public void onWeightChanged(String weightInfo) {
    runOnUiThread(() -> updateUIWeight(weightInfo));
}
```

4. 正常业务功能：

在收到服务连接建立成功的回调函数中，APP 根据业务需要做初始化或调用其它 API 接口，主动实现某些业务功能：

例 1：设置皮重：

```
@Override
public void onMtWeighServiceConnected() {
    doSomeInit();
    mMtWeightService.setTareAsync();
}
```

例 2：打印文本：

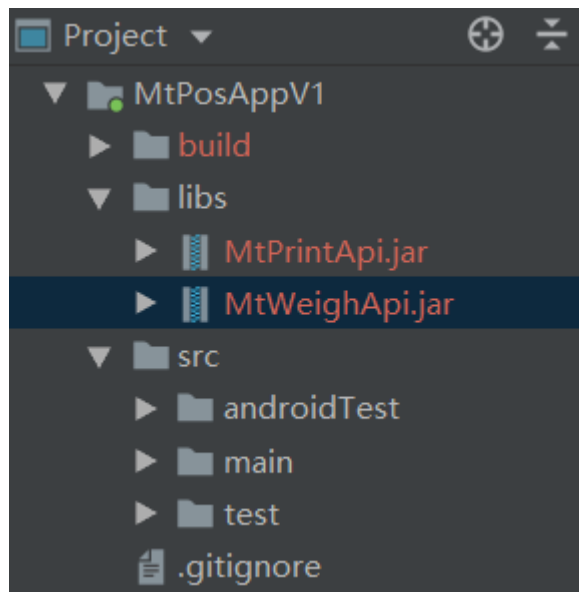
```
@Override
public void onMtPrintServiceConnected() {
    doSomeInit();
    MtPrintResult result = new MtPrintResult();
    if (mMtPrintService.getPrinterStatus(result)) {
        mMtPrintService.printText("Hello World.");
    }
}
```

2.5 文件清单

文件名称	说明
MtPrintApi.jar	MT 提供的打印服务接口 API 文件，其为 AIDL 文件和相关类的包装库，目的是方便其它 APP 调用和保证 API 版本的兼容性；
MtWeighApi.jar	和 MtPrintApi.jar 类似，是针对称重服务的接口 API；

2.6 安装路径

上述[文件清单](#)中提到的文件由梅特勒-托利多公司提供，统一放在 APP 项目的/libs 目录下，如下：



然后点击 Android Studio 的工具栏上图标或菜单 File->Sync Project with Gradle Files，同步项目文件。

2.7 ADB 调试

支持两种联接方式：Mirco USB 和 ADB 网络；

2.7.1 Micro USB 调试



2.7.2 网络调试

有线网络相对无线网络传输稳定快速，推荐开发者使用有线网络方式。

办公网络有 DHCP 服务器的，插上以太网即可获取到 IP 地址，否则需要手动设置静态 IP。

➤ Adb 连接到秤

在 Android studio 的 Terminal 标签下输入 ADB 命令: `adb connect IP:5555`

```
Terminal: Local x +
Microsoft Windows [版本 10.0.17763.1039]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\02_Code\04_FreshAce_2019>adb connect 192.168.1.40:5555
connected to 192.168.1.40:5555

C:\02_Code\04_FreshAce_2019>
```

➤ 查看有线 IP 地址

设置=》无线和网络=》更多=》以太网

保密信息

梅特勒-托利多(常州)测量技术有限公司
Mettler-Toledo (Changzhou) Measurement Technology Ltd.

地址: 江苏省常州市新北区
太湖西路111号
电话: 86-0519-86642040
传真: 86-0519-86641991
邮编: 213125

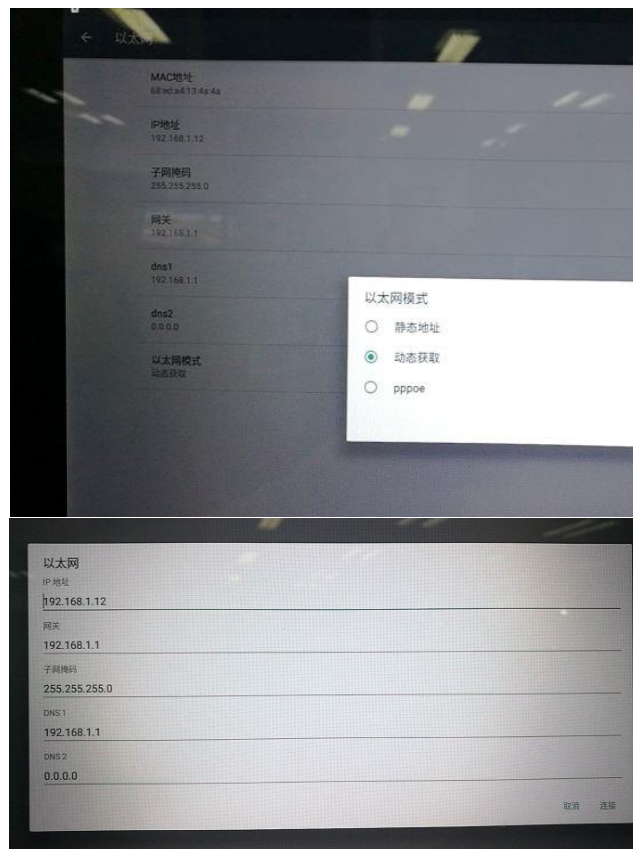
Address: 111 Taihu West Road,
Changzhou, Jiangsu
Tel: 86-0519-86642040
Fax: 86-0519-86641991
Zip: 213125

www.mt.com



➤ 设置静态 IP

设置=》无线和网络=》更多=》以太网=》以太网模式=静态地址



➤ 查看无线 IP 地址

设置=》系统=》关于设备=》状态信息=》IP 地址



3. 称重接口

3.1 概述

Android 应用属于典型的事件驱动类型，异步通知设计更符合 Android 的风格，但是，考虑到 MT 的 API 数量较多，如果都采用异步则需要定义很多回调接口，对于 APP 来说实现起来就复杂了，因此，MT 只对那些阻塞时间长的 API 提供异步回调接口；同时，API 中也实现了相同功能的同步阻塞函数，APP 可以在异步函数不方便使用的场景使用，但只能在非 UI 线程中使用，否则很可能会 ANR；

称重服务中如秤盘不稳定，则可能会导致函数阻塞时间较长：包括：称量皮重、预置皮重和清零功能；

3.2 获取称重 API 实例

称重 Api 的类为 MtWeighApi，通过 getInstance() 获取其单实例对象，所有称重相关接口均为该类的成员函数，下面依次介绍这些公共成员函数。

例：MtWeighApi.getInstance();

3.3 建立连接

APP 使用称重服务前，需要先建立到称重服务的通讯连接。

【函数原型】	boolean connectToService(Context context, IMtWeighView iMtWeighView) void disconnectService(Context context)
【功能说明】	分别实现发起到称重服务的 Binder 连接请求和断开连接请求；

【参数说明】	context: 当前 APP 的上下文, 内部使用的是 Application 的 Context, 目的是在 APP 运行情况下, 始终保持到称重服务的 Binder 连接; IMtWeighView: 称重服务回调接口类, APP 需要实现该接口类, 包括: 连接建立成功回调、连接中断回调、重量变化通知回调、基本信息变化回调、设置皮重结束回调和清零操作结束回调, 详见下面章节描述。
【返回值】	true: 连接建立成功; false: 连接建立失败
【使用方法】	APP 必须在调用其它功能函数之前调用本函数建立通讯连接, 且同时设置好回调接口

3.4 回调接口说明

称重服务声明了回调接口 IMtWeighView, 其包括两类函数: 1. 连接建立和中断回调; 2. 数据或状态发生改变回调; 其中, 设置连接建立和中断的回调接口原因是 Android 系统自身机制导致: APP 每次启动新 Activity 或 Service 组件都只是发送启动请求给系统, 之后由系统异步启动目标组件, 最后系统通过回调接口通知 App 新连接建立成功, 因此, APP 必须实现该回调; 其它接口是因为 Model 驱动刷新显示 UI 所需要;

```
public interface IMtWeighView {
    void onMtWeighServiceConnected();
    void onMtWeighServiceDisconnected();
    void onWeightChanged(String msg);
    void onBaseInfoChanged(String msg);
    void onSetTareFinished(String msg);
    void onZeroFinished(int result);
}
```

3.4.1 连接状态通知

当 onMtWeighServiceConnected()函数被调用时, 表示 APP 到 MtWeighService 的通讯连接建立成功, APP 可以在本函数中完成显示重量相关 UI 的初始化工作。

当 onMtWeighServiceDisconnected()被调用时表示 APP 到 MtWeighService 的通讯连接中断; APP 可在本函数中尝试再次发起连接, 或触发 UI 显示不能称重的错误信息。

3.4.2 更新重量

APP 和称重服务成功建立连接后, 当秤盘上重量发生变化时, 服务会调用函数 onWeightChanged()通知新的重量信息, APP 可在本函数中不断刷新显示 UI 上的重量信息;

【函数原型】	void onWeightChanged(String msg)
【功能说明】	刷新 APP 的 UI 视图上关于重量信息或报错;

【参数说明】	msg: 新的重量信息或错误信息, JSON Object 格式, 详见 Json 参数各 Key 含义; 例: {"result":"0","net":"0.028","zero":"0","tareType":"N","tare":"0.000","unit":"kg"}
【返回值】	无
【使用方法】	<ul style="list-style-type: none"> ◆ 商用衡器通常都是用净重作为显示重量, 当设置了皮重时, 需要在用户界面上明确显示皮重类型和皮重值。 ◆ 当 result 值非 0 时, 表示结果异常, 显示界面上需要根据具体错误原因显示不同的内容, 比如重量欠载或超重时, 在用户界面上显示相应的符号或文字。

Json 参数各 Key 含义:

字段	含义
result	本次操作执行结果, 0 表示成功; -1000 表示不稳定重量, 不能作为收银结果; -1001 表示欠载; -1002 表示超重; -1003 表示无法取重; 其它值表示失败, App UI 上需要分别处理这里提到的几种情况。返回码详见 返回码及含义 ;
unit	重量单位, 如 kg、g 等, 当前只支持 kg;
net	净重;
zero	表明是否在毛重零位, 1: 在 0 位, 0: 不是 0 位;
tareType	皮重类型: N: 没皮重; PT: 预置皮重; T: 测量皮重。根据法规, 如果使用了皮重, 则必须在重量显示界面显示皮重类型;
tare	当前皮重值

3.4.3 更新基本信息

【函数原型】	void onBaseInfoChanged (String msg)
【功能说明】	刷新 APP UI 上关于最大量程、最小量程、精度信息、最大预置皮重等信息;
【参数说明】	msg: 量程等基本信息, JSON Object 格式; 例: {"result":"0","minRange":"0.040","maxRange1":"6.000","resolution1":"0.002","unit1":"kg","rangeNumber":"2","maxRange2":"15.000","resolution2":"0.005","unit2":"kg","minPresetTare":"0.000","maxPresetTare":"6.000","tareUnit":"kg"}
【返回值】	无
【使用方法】	APP 实现本接口函数时, 可在本函数中刷新 UI 页面的基本信息

Json 参数各 Key 含义:

字段	含义
result	本次操作执行结果, 0 表示成功, 其它值表示异常或失败, 详见 返回码及含义
rangeNumber	当前秤是几量程, 如单量程为 1, 双量程为 2;
maxRange1	第一分度最大量程
maxRange2	第二分度最大量程
maxRange3	第三分度最大量程, 预留
minRange	最小量程, 为 20*检测分度值
resolution1	第一分度检定分度值
resolution2	第二分度检定分度值
resolution3	第三分度检定分度值, 预留
unit1	第一量程单位, 默认 kg
unit2	第二量程单位, 默认 kg
unit3	第三量程单位, 默认 kg
minPresetTare	最小预置皮重值
maxPresetTare	最大预置皮重值
tareUnit	预置皮重单位

3.4.4 皮重设置完成通知

【函数原型】	void onSetTareFinished(String msg)
【功能说明】	通知 APP 设置测量皮重或预置皮重结束, 需配合 异步设置测量皮重 和 异步设置预置皮重 接口使用;
【参数说明】	msg: 皮重结果信息, JSON Object 格式; 例如: {"result": "0", "tare": "1.000", "unit": "kg"} 注意: 预置皮重场景, 生效的皮重值可能和配置的不同, 因为称重平台会根据精度进行自动圆整。
【返回值】	无
【使用方法】	APP 在 UI 上显示的皮重值应取自 onWeightChanged 中的皮重值和类型, 而不是用本函数的返回值;

Json 参数各 Key 含义:

字段	含义
result	本次操作执行结果。0 表示成功, 其它值表示失败, 详见 返回码及含义
tare	设置成功后的皮重值
unit	皮重单位

3.4.5 清零完成通知

【函数原型】	void onZeroFinished (int result)
【功能说明】	通知 APP 执行清零操作结束, 需配合 异步清零接口 使用;
【参数说明】	result: 异步清零操作结果, 0 表示成功, 其它失败, 详见 返回码及含义 。
【返回值】	无
【使用方法】	APP 实现本接口函数时, 可设置 UI 页面显示效果或不处理;

3.5 设置称量皮重

3.5.1 同步接口

【函数原型】	String setTare ()
【功能说明】	同步阻塞方式设置测量皮重, 根据秤盘上当前重量进行自动皮重设置;
【参数说明】	无
【返回值】	返回结果的 JSON 字符串, 包括当前皮重值; 例: {"result": "0", "tare": "0.014", "unit": "kg"}
【使用方法】	<ul style="list-style-type: none"> ➤ 如果去皮时秤盘上重量不稳定, 可能需要等待稳定后才能去皮成功; 如超过一定时间重量仍然无法稳定, 则去皮动作可能失败。 ➤ 根据法规, 在设置了皮重后, 应用显示界面需要明确显示称重皮重标记 T, 详见onWeightChanged()说明。

Json 参数各 Key 含义:

字段	含义
----	----

Result	本次操作执行结果。0 表示成功，其它值表示失败，详见 返回码及含义
tare	设置成功后的皮重值
unit	皮重单位

3.5.2 异步接口

【函数原型】	int setTareAsync()
【功能说明】	非阻塞的设置测量皮重，下发设置皮重命令后立即返回，不等待执行结果，结果通过 <code>ImtWeighView</code> 的 <code>onSetTareFinished(String msg)</code> 异步通知；
【参数说明】	无
【返回值】	返回 int 类型返回码，详见 返回码及含义
【使用方法】	参见同步接口说明

3.6 设置预置皮重

3.6.1 同步接口

【函数原型】	String setPresetTare (String strTareValue)
【功能说明】	同步阻塞方式设置预置皮重，根据传入的皮重值参数进行去皮；
【参数说明】	<ul style="list-style-type: none"> ➤ <code>strTareValue</code>: 入参: 表示要设置的皮重值，单位: kg; 字符串格式（非 JSON），如: “0.128”； ➤ 根据值不同有两个不同含义: 1. 假如值为字符串 0 (“0” 或 “0.000”)则表示清除皮重值; 2. 其他非“0”时表示要设置的皮重值; ➤ 建议设置的皮重值为小数点后 3 位浮点数，如: “0.002”。
【返回值】	返回结果的 JSON 字符串，包括当前生效的皮重值； 例: {"result": "0", "tare": "0.014", "unit": "kg"}
【使用方法】	<ul style="list-style-type: none"> ➤ 本函数用于某些需要去除包装重量的场景； ➤ 根据法规，在设置了皮重后，应用显示界面需要明确显示预置皮重 PT 标记和皮重值，详见 onWeightChanged() 函数的 <code>tareType</code> 说明。 注意: 皮重值建议按照最小检测分度值整数倍设置，否则，称重平台会根据秤的精度进行圆整操作，而使最终皮重值和设置皮重值不同；

Json 参数各 Key 含义:

字段	含义
result	本次操作执行结果。0 表示成功，其它值表示失败，详见 返回码及含义
tare	设置成功后的皮重值
unit	皮重单位

3.6.2 异步接口

【函数原型】	int setPresetTareAsync(String strTareValue)
【功能说明】	非阻塞的设置预置皮重，下发设置皮重命令后立即返回，不等待执行结果，结果通过 <code>ImtWeighView</code> 的 <code>onSetTareFinished(String msg)</code> 异步通知；
【参数说明】	参考 同步阻塞接口 中描述
【返回值】	返回 int 类型返回码，详见 返回码及含义
【使用方法】	参考 同步阻塞接口 中描述。

3.7 清除皮重

【函数原型】	int clearTare ()
【功能说明】	清除之前设置的测量皮重或预置皮重值；
【参数说明】	无
【返回值】	操作结果的返回码，详见 返回码及含义
【使用方法】	<ul style="list-style-type: none"> ➤ 毛重零时（卸掉所有载荷后的空秤台），才允许清除测量皮重。 ➤ 测量皮重和预置皮重都能够通过本接口清除。

3.8 手动清零

3.8.1 同步接口

【函数原型】	int setZero (void)
【功能说明】	清零（同步阻塞）功能，因为有灰尘或其他原因导致秤面显示重量非 0，如：-0.002 或 0.004，此时可用本接口将重量快速设置为 0；

【参数说明】	无
【返回值】	操作结果, 0 表示成功, 其它失败, 详见 返回码及含义
【使用方法】	<ul style="list-style-type: none"> ➢ 因为有灰尘或其他原因导致空秤台显示重量非 0 时使用本接口; ➢ 当前重量需要在清零范围内 (满量程+- 2%) 才能清零成功。以满量程为 15kg 的秤为例, 清零范围: -300g ~ 300g;

3.8.2 异步接口

【函数原型】	int setZeroAsync()
【功能说明】	非阻塞的执行清零操作, 下发清零命令后立即返回, 不等待执行结果, 结果通过 lmtWeighView 的 onZeroFinished(int retCode) 异步通知
【参数说明】	无
【返回值】	操作结果, 0 表示成功, 其它失败, 详见 返回码及含义
【使用方法】	参考同步接口使用方法说明;

4. 打印接口

4.1 调用打印机说明

打印服务设计可同时支持多种不同类型打印机, 用于分别打印收据、标签或无基纸, 因此, 在多打印机场景下, API 参数中需要指定打印机类型字符串 label_2inch, ticket_2inch 或 linerless_2inch, 用以区分要使用的打印机。

对于单打印机场景, 为了简化 APP 代码, MT 的打印 API 中也包含了不需要打印机类型参数的接口函数, 其功能和需要指定打印机类型参数的完全相同;

一次票据打印开始前, 建议先用 [getPrinterStatus](#) 接口查询打印机状态是否为可开始打印的空闲态, 防止打印机不存在、缺纸等错误情况。

4.2 获取称重 API 实例

打印 Api 的类为 MtPrintApi, 通过 getInstance() 获取其单实例对象, 所有打印相关接口均为该类的成员函数, 下面依次介绍这些公共成员函数

例如: `MtPrintApi.getInstance();`

4.3 建立连接

APP 使用打印服务前，需要先建立到服务的通讯连接。

【函数原型】	boolean connectToService(Context context, IMtPrintView iMtPrintView) void disconnectService(Context context)
【功能说明】	分别实现发起到服务的 Binder 连接请求和断开连接请求；
【参数说明】	context: 当前 APP 的上下文，内部使用的是 Application 的 Context，目的是在 APP 运行情况下，始终保持到服务的 Binder 连接； iMtPrintView: 打印服务回调接口，包括：连接建立成功回调、连接中断回调、打印机列表更新通知回调、FeedPaper 结束回调和打印结束回调，详见下面章节描述。
【返回值】	true: 连接建立成功； false: 连接建立失败；
【使用方法】	APP 必须在调用其它功能函数之前调用本函数建立通讯连接，且同时设置好回调接口；

4.4 回调接口说明

打印服务异步回调接口类 IMtPrintView，APP 在同服务通信前必须实现并注册本接口，以接收异步事件的通知。

```
public interface IMtPrintView {
    void onMtPrintServiceConnected();
    void onMtPrintServiceDisconnected();
    void onPlugEvent(String printer, int iEvent);
    void onPrinterListChanged(List<String> printerList);
    void onFeedPaperFinished(int retCode);
    void onPrintFinished(int retCode);
}
```

4.4.1 连接状态通知

当 onMtPrintServiceConnected()函数被调用时，表示 APP 到 MtPrintService 的通讯连接建立成功，APP 可以在本函数中继续触发依赖 MTPrintService 的组件的初始化工作。

当 onMtPrintServiceDisconnected ()被调用时表示 APP 到 MtPrintService 的通讯连接中断；APP 可在本函数中可再次尝试建立连接，或触发 UI 显示打印连接断开的错误信息，以便获得较好的用户体验。

4.4.2 热插拔事件通知

【函数原型】	void onPlugEvent(String printer, int iEvent);
【功能说明】	打印服务通过本回调通知 USB 打印机热插拔事件；
【参数说明】	printer: 发生事件的打印机类型；目前可能为：ticket_2inch 或 linerless_2inch iEvent: 0: 打印机 printer 被拔出；1: 打印机 printer 被插入；
【返回值】	无
【使用方法】	USB 打印机支持热插拔功能，因此，会通知 UI 拔插事件；本事件通知早于打印机列表变化通知 onPrinterListChanged 。

4.4.3 打印机列表变化

【函数原型】	void onPrinterListChanged(ArrayList<String> printerList)
【功能说明】	打印服务通过本回调通知可用打印机列表
【参数说明】	printerList: 可用打印机类型的 ArrayList 类型列表；
【返回值】	无
【使用方法】	当应用 APP 显示打印机列表时可以得到实时打印机列表变化通知。

4.4.4 走纸结束通知

【函数原型】	void onFeedPaperFinished(int retCode)
【功能说明】	当使用 feedPaperAsync() 执行异步走纸功能时，MtPrintService 通过本回调通知打印机走纸结束事件；
【参数说明】	retCode: 命令执行结果的返回值；0: 表示正常结束，其它：操作失败，详见 返回码及含义 。
【返回值】	无
【使用方法】	只有 APP 用到走纸功能（异步）时，才需要应用到本函数；

4.4.5 打印结束通知

【函数原型】	void onPrintFinished(int retCode)
【功能说明】	当使用打印文本、图片结束时，MtPrintService 通过本回调通知打印结束；

【参数说明】	retCode: 命令执行结果的返回值; 0: 表示正常结束, 其它: 操作失败, 详见 返回码及含义 。
【返回值】	无
【使用方法】	<ul style="list-style-type: none"> ➤ 在每笔交易必须确保打印完成场景时, 需要检查本函数返回值; 举例说明: 在打印过程如果发生缺纸, 则实际数据是未打印成功的, 此时, 本函数会分返回缺纸错误, APP 如果不检查返回值, 则不知道发生缺纸或其它错误; ➤ 当连续调用打印文本或图片 API 间隔小于 200ms 时, 会认为是同一次打印, 只在最后执行一次本函数, 而不是每行打印都执行本函数!

4.4.6 出参类型说明

打印服务接口函数较多, 但从数据流方向上看基本都是发送命令和数据到打印机, 从打印机读取的数据很少, 因此, 设计不同于称重服务的 JSON 格式返回方案, 而是采用了自定义类 MtPrintResult 作为出参。这样方便在命令执行失败时查看错误码, 以及在查询数据时, 将结果字符串返回给 APP。

MtPrintResult 类两个关键成员函数如下:

```
public class MtPrintResult implements Parcelable {
    private int mReturnCode;
    private String mMsg;
    public int getReturnCode(){
        return this.mReturnCode;
    }
    public String getMsg(){
        return this.mMsg;
    }
    ...
}
```

注意:

1. 客户端只有在接口函数返回 false 的情况下, 才需要查看本类内容, 以确认失败原因;
2. 因为大部分错误信息可以通过错误码表示, 因此, getMsg 返回的字符串大部分情况为空;
3. 只有在查询打印机某些内容时 (如: 固件版本), 才会使用 getMsg 查看字符串内容;

4.5 查询打印机状态

【函数原型】	boolean getPrinterStatus(MtPrintResult result) boolean getPrinterStatus(String printerType, MtPrintResult result)
--------	--

【功能说明】	查询打印机当前的状态;
【参数说明】	printerType: 入参, 打印机类型; 多打印机场景使用; result: 出参, 用于失败时通知 APP 错误码, 表示获取失败或状态异常 (包括打印机忙), 其它失败原因详见 返回码及含义 ;
【返回值】	true: 则表示打印机空闲, 可开始打印; false: 命令执行失败或打印机状态异常;
【使用方法】	APP 打印新票据前, 先调用本函数查看打印机状态, 如打印机状态空闲, 则开始打印文本或图片; 如发现打印机状态出错, 则可在 UI 上提示打印机出错信息;

4.6 打印文本

【函数原型】	boolean printText(String strText, MtPrintResult result) boolean printText(String printerType, String strText, MtPrintResult result) boolean printTextWithFont(String strText, int iFontSize, MtPrintResult result) boolean printTextWithFont(String printerType, String strText, int iFontSize, MtPrintResult result) boolean printTextWithFontAndAlignment(String strText, int iFontSize, int iAlign, MtPrintResult result) boolean printTextWithFontAndAlignment(String printerType, String strText, int iFontSize, int iAlign, MtPrintResult result)
【功能说明】	将字符串 strText 按照指定字体大小和对齐方式用指定名称的打印机打印出来, 如未指定字体、对齐方式和打印机类型, 则使用默认值.;
【参数说明】	strText: 待打印字符串, UTF-8 编码, 支持\n 分隔的多行文字; printerType: 入参, 打印机类型; 多打印机场景使用; iFontSize: 字号的像素高度, 必须在[12, 160]区间, 单位 px; 修改默认字体大小参见函数 setDefaultFontSize ; iAlign: 该字符串格式化打印的对齐方式, 0: 左对齐; 1: 居中; 2: 右对齐。修改默认值参见函数 setDefaultAlignment ; result: 出参, 用于失败时通知 APP 错误码, 详见 返回码及含义
【返回值】	true: 命令执行成功; false: 命令执行失败;
【使用方法】	<ul style="list-style-type: none"> ➢ printText 以默认纸张宽度进行打印, 默认等于打印头宽度, 除非使用 setPaperWidth 修改了默认纸张宽度; ➢ 当打印收据时, 可连续调用本函数, 每次打印一行; ➢ 如字符串格式化后超过打印一行, 则默认自动换行继续打印; <p>注意: 一次性传入的文本太多可能导致打印开始前的等待时间变长, 因为需要做文本格式化。</p>

4.7 打印多列文本

【函数原型】	<pre>boolean printColumnsText(String[] columnsText, int[] columnsWidth, int[] columnsAlign, MtPrintResult result) boolean printColumnsText(String printerType, String[] columnsText, int[] columnsWidth, int[] columnsAlign, MtPrintResult result)</pre>
【功能说明】	按指定相对比例宽度打印几列文本在一行上；
【参数说明】	<p>printerType: 入参，打印机类型，多打印机场景使用；</p> <p>columnsText 待打印的各列字符串数组</p> <p>columnsWidth 待打印的各列字符串的宽度权重值数组 单位：权重值</p> <p>如：共两列文本，其权重值都为 1，则表示各占 50%</p> <p>columnsAlign 待打印的各列字符串的对齐方式 0：左对齐； 1：居中； 2：右对齐</p> <p>备注：三个数组参数长度必须一致；</p> <p>当 columnsText[i] 的宽度大于 columnsWidth[i] 时，文本会换行</p> <p>result: 出参，用于失败时通知 APP 错误码，详见返回码及含义；</p>
【返回值】	<p>true: 命令执行成功；</p> <p>false: 命令执行失败；</p>
【使用方法】	打印商品流水时使用，每行可按名称、单价、重量、小计多列打印。

4.8 打印位图

【函数原型】	<pre>boolean printBitmap(Bitmap bitmap, MtPrintResult result) boolean printBitmap(String printerType, Bitmap bitmap, MtPrintResult result) boolean printBitmap(Bitmap bitmap, int align, int offset, MtPrintResult result) boolean printBitmap(String printerType, Bitmap bitmap, int align, int offset, MtPrintResult result)</pre>
【功能说明】	<p>将 Bitmap 原始位图按照对齐方式和偏移量进行打印；</p> <p>如位图原始宽大于打印纸宽度，则自动截断；</p> <p>当不指定对齐方式和偏移量时，默认图片左对齐，偏移量为默认偏移值；</p>
【参数说明】	<p>bitmap: 入参，位图数据；需要 APP 将图片转换为 Bitmap 格式后传入；</p> <p>printerType: 入参，打印机类型，多打印机场景使用；</p> <p>align: 入参，图片在打印纸上的对齐方式，0：左对齐； 1：居中； 2：右对齐；</p> <p>offset: 入参，正整数，左或右对齐时，图片打印位置偏离对齐边的点数；居中时本参</p>

	数无效; result: 出参, 用于失败时通知 APP 错误码, 详见 返回码及含义 ;
【返回值】	true: 命令执行成功; false: 命令执行失败;
【使用方法】	在纸上打印 Logo, 二维码和打印标签。

4.9 打印空白行

【函数原型】	boolean printBlankLine(int iLines, MtPrintResult result) boolean printBlankLine(String printerType, int iLines, MtPrintResult result)
【功能说明】	打印若干空行, 行高为按照打印机默认字体计算的高度;
【参数说明】	iLines: 入参, 打印空白行数, 单位: 文本行数; printerType: 入参, 打印机类型, 多打印机场景使用; result: 出参, 用于失败时通知 APP 错误码, 详见 返回码及含义
【返回值】	true: 命令执行成功; false: 命令执行失败;
【使用方法】	打印多行文本中间需要增加空行时使用;

4.10 开始打印

【函数原型】	void beginPrint(String strPrinterType)
【功能说明】	立即开始打印缓冲区的文本, 主要针对标签打印场景。
【参数说明】	strPrinterType: 入参, 打印机类型, 多打印机场景使用;
【返回值】	无
【使用方法】	对于打印标签, 推荐 POS 软件通过 printBitmap(Bitmap bitmap) 一次将整个格式化好的标签 Bitmap 传给打印服务, 而不是多次调用 printText 或 printBitmap 系列函数, 好处是可以立即开始打印; 否则, 部分款型标签打印机需在缓存超时 (200 ~ 300ms) 后开始打印; 本函数正是为了解决这个缓存导致延迟打印问题, 其显示通知打印服务缓存数据结束, 立即开始打印。

4.11 设置默认值

4.11.1 设置默认对齐方式

【函数原型】	boolean setDefaultAlignment(int iAlignment, MtPrintResult result) boolean setDefaultAlignment(String printerType, int iAlignment, MtPrintResult result)
【功能说明】	设置默认打印对齐方式，用于 printText 不指定打印对齐方式参数时的默认值；
【参数说明】	iAlignment: 对齐方式，0：左对齐；1：居中；2：右对齐； printerType: 入参，打印机类型；多打印机场景使用； result: 出参，用于失败时通知 APP 错误码，返回值详见 返回码及含义
【返回值】	true: 命令执行成功； false: 命令执行失败；
【使用方法】	打印文本时，默认居中或居右对齐时使用。

4.11.2 设置默认字体大小

【函数原型】	boolean setDefaultFontSize(int iSize, MtPrintResult result) boolean setDefaultFontSize(String printerType, int iSize, MtPrintResult result)
【功能说明】	设置默认打印文本字体大小，用于 printText 不指定字体大小参数时的默认值； 打印空白行高也受本函数影响；
【参数说明】	iFontSize: 字号的大小，需要 APP 根据实际打印效果设置；范围：[12, 160]； printerType: 入参，打印机类型；多打印机场景使用； result: 出参，用于失败时通知 APP 错误码，返回值详见 返回码及含义 ；
【返回值】	true: 命令执行成功； false: 命令执行失败；
【使用方法】	调用打印文本接口时，且打印字体大小基本都一致时使用； printBlankLine 时默认字体尺寸的高度。

4.11.3 设置默认左边距

【函数原型】	boolean setLeftMargin(int marginPixel, MtPrintResult result) boolean setLeftMargin(String printerType, int marginPixel, MtPrintResult result)
【功能说明】	设置默认左边距值；
【参数说明】	marginPixel: 边距值，正整数 单位：px；

	printerType: 入参, 打印机类型; 多打印机场景使用; result: 出参, 用于失败时通知 APP 错误码, 返回值详见 返回码及含义 ;
【返回值】	true: 命令执行成功; false: 命令执行失败;
【使用方法】	用于打印起始位置太靠近左边界或出了边界时 微调 打印起始位置;

4.11.4 设置默认右边距

【函数原型】	boolean setRightMargin(int marginPixel, MtPrintResult result) boolean setRightMargin(String printerType, int marginPixel, MtPrintResult result)
【功能说明】	设置默认右边距值;
【参数说明】	marginPixel: 边距值, 正整数 单位: px; printerType: 入参, 打印机类型; 多打印机场景使用; result: 出参, 用于失败时通知 APP 错误码, 返回值详见 返回码及含义 ;
【返回值】	true: 命令执行成功; false: 命令执行失败;
【使用方法】	用于打印文本右侧太靠近右边界或出了边界时 微调 打印位置;

4.11.5 设置打印纸宽度

【函数原型】	boolean setPaperWidth(int iWidth, MtPrintResult result) boolean setPaperWidth(String printerType, int iWidth, MtPrintResult result)
【功能说明】	设置默认打印纸宽度, 小于等于打印头宽度;
【参数说明】	iWidth: 打印纸默认宽度, 单位: mm; printerType: 入参, 打印机类型; 多打印机场景使用; result: 出参, 用于失败时通知 APP 错误码, 返回值详见 返回码及含义 ;
【返回值】	True: 命令执行成功; False: 命令执行失败;
【使用方法】	当打印纸宽度小于打印头宽度不同时需要设定本参数, 否则可能打印起始位置出错, 导致打印内容缺失。

4.11.6 设置字体缩放系数

【函数原型】	boolean setScaleX(float scaleX, MtPrintResult result)
--------	---

	boolean setScaleX(String printerType, float scaleX, MtPrintResult result)
【功能说明】	设置默认打印字沿 X 轴（即：宽度）缩放系数，以达到一行内打印更多字符目的。缩放系数默认值 1，小于 1 字体变窄，大于 1 变宽；
【参数说明】	scaleX: 打印字体沿 X 轴缩放系数，取值范围: [0.3, 2]，默认值 1，建议值 1 和 0.5； printerType: 入参，打印机类型；多打印机场景使用； result: 出参，用于失败时通知 APP 错误码，返回值详见 返回码及含义
【返回值】	True: 命令执行成功； False: 命令执行失败；
【使用方法】	当每行内容较多放不下，又不希望换行打印时，可调用本接口使打印字更窄，容纳更多字符。 注意：修改字号是整个字体变大/变小，而本接口只在 X 轴方向缩放，且只对打印文字生效，对图片、条码不生效；

4.12 辅助功能

4.12.1 打印分隔线

【函数原型】	boolean printSeparationLine(String separator, MtPrintResult result) boolean printSeparationLine(String printerType, String separator, MtPrintResult result)
【功能说明】	打印一行分割线，行高为按照打印机默认字体计算的高度；
【参数说明】	printerType: 入参，打印机类型，多打印机场景使用； separator: 入参，分行符号，如：“-”或“=”； 如分隔字符数超出一行则自动截断为一行； 不支持使用超过一行的混排字符串做分隔符； result: 出参，用于失败时通知 APP 错误码，详见 返回码及含义
【返回值】	true: 命令执行成功； false: 命令执行失败；
【使用方法】	打印收据时多项商品之间、标题和流水之间可以用分割线分割，显示效果会好些；

4.12.2 打印一维条码

【函数原型】	boolean printBarCode(String strText, int iFormat, int width, int height, int align, int offset, int textPosition, MtPrintResult result) boolean printBarCode(String printerType, String strText, int iFormat, int width, int
--------	---

	height, int align, int offset, int textPosition, MtPrintResult result)
【功能说明】	打印一维条码;
【参数说明】	<p>printerType: 入参, 打印机类型, 多打印机场景使用;</p> <p>strText: 入参, 条码的内容; 不同的条码类型支持的最大打印字符数和支持的字符范围可能不同;</p> <p>iFormat: 一维码类型, 取值:</p> <ul style="list-style-type: none"> 1--UPC-E, 2--JAN13(EAN13), 3--JAN8(EAN8), 4--CODE39, 5--ITF, 6--CODABAR, 7--CODE93, 8--CODE128 <p>width: 条码宽度, 单位: px</p> <p>height: 条码高度, 单位: px</p> <p>align: 条码对齐方式, 取值范围:</p> <ul style="list-style-type: none"> 0: 居左; 1: 居中; 2: 居右 <p>offset: 在对齐方式为居左或居右时, 偏移边的距离, 单位: px;</p> <p>textPosition: 打印条码内容相对条码图的位置, 取值范围:</p> <ul style="list-style-type: none"> 0: 不打印; 1: 文字在条码上方; 2: 文字在条码下方; <p>注: 条码内容文字大小为默认字体的一半;</p> <p>result: 出参, 用于失败时通知 APP 错误码, 详见返回码及含义</p>
【返回值】	<p>true: 命令执行成功;</p> <p>false: 命令执行失败;</p>
【使用方法】	需要打印一维条码时使用;

4.12.3 打印二维码

【函数原型】	boolean printQRCode(String strText, int width, int align, int offset, MtPrintResult
--------	---

	result) boolean printQRCode(String printerType, String strText, int width, int align, int offset, MtPrintResult result)
【功能说明】	打印二维码;
【参数说明】	printerType: 打印机类型, 多打印机场景使用; strText: 条码的内容; width: 条码宽度, 单位: px align: 条码对齐方式, 取值范围: 0: 居左; 1: 居中; 2: 居右 offset: 正整数, 在对齐方式为居左或居右时, 偏移边的距离, 单位: px; result: 出参, 用于失败时通知 APP 错误码, 详见 返回码及含义
【返回值】	true: 命令执行成功; false: 命令执行失败;
【使用方法】	打印简单二维码场景下使用; 如希望打印带有特殊图片的二维码, 建议 APP 自行实现, 然后调用 printBitmap 接口打印;

4.12.4 切纸

【函数原型】	boolean cutPaper(int iPercent, MtPrintResult result) boolean cutPaper(String printerType, int iPercent, MtPrintResult result)
【功能说明】	执行切刀切纸操作, 不是所有打印机都支持本功能;
【参数说明】	iPercent: 表示切纸的百分比, 范围[1, 100]; printerType: 打印机类型; 用于多打印机场景; result: 出参, 用于失败时通知 APP 错误码, 详见 返回码及含义
【返回值】	true: 命令执行成功; false: 命令执行失败;
【使用方法】	打印收据结束时, 用切刀切断收据纸, 方便客户拿收据;

4.12.5 走纸

同步接口:

【函数原型】	boolean feedPaper(int distance_mm, MtPrintResult result) boolean feedPaper(String printerType, int distance_mm, MtPrintResult result)
【功能说明】	以同步阻塞方式控制打印机走纸指定的距离。
【参数说明】	printerType: 入参, 打印机类型; 多打印机场景使用; distance_mm: 入参, 打印机走纸距离, 单位: 毫米; result: 出参, 用于失败时通知 APP 错误码, 返回值详见 返回码及含义
【返回值】	true: 命令执行成功; false: 命令执行失败;
【使用方法】	<ul style="list-style-type: none"> ➤ 一般用于标签打印机更换标签纸后走一次纸, 以对齐标签纸缝隙; ➤ 当打印机不支持硬件走纸, 需要 APP 软件实现走纸功能可以使用本接口; ➤ 不要在打印文本过程中调用本函数走纸, 这种场景使用 printBlankLine 函数。

异步走纸:

【函数原型】	boolean feedPaperAsync (int distance_mm, MtPrintResult result) boolean feedPaperAsync (String printerType, int distance_mm, MtPrintResult result)
【功能说明】	异步触发打印机走纸指定参数的距离, 配合 onFeedPaperFinished 回调使用。
【参数说明】	printerType: 打印机类型; 多打印机场景使用; distance_mm: 打印机走纸距离, 单位: 毫米; result: 出参, 用于失败时通知 APP 错误码, 返回值详见 返回码及含义
【返回值】	true: 命令执行成功; false: 命令执行失败;
【使用方法】	参考 feedPaper

4.12.6 获取打印机列表

【函数原型】	ArrayList<String> getPrinterList()
【功能说明】	多打印机场景, 读取打印机列表;
【参数说明】	无
【返回值】	打印机类型 ArrayList, 当 List 长度为 0 时表示无打印机或打印机未初始化成功;
【使用方法】	<ul style="list-style-type: none"> ➤ 多打印机场景, APP 需要指定使用哪个打印机打印; ➤ 单打印机场景: 不用关注本函数;

4.12.7 查询 API 版本

【函数原型】	String getApiInfo()
【功能说明】	获取本 Print Service 的 API 版本等信息；
【参数说明】	printerType: 打印机类型；多打印机场景使用； result: 出参，用于通知 APP 打印结果成功或失败，成功时通过 getMsg()成员函数读取版本信息；出错时返回值详见 返回码及含义
【返回值】	API 版本信息字符串，例如：Print_API_Ver: 1.0.0；
【使用方法】	维护用；

4.13 钱箱相关

4.13.1 打开钱箱

【函数原型】	int openCashDrawer()
【功能说明】	打开钱箱。
【参数说明】	无
【返回值】	返回命令执行结果，详见 返回码及含义 ；
【使用方法】	无

5. 系统接口

5.1 沉浸式接口说明

Android 系统提供了一些控制接口，以方便一些未实现沉浸式 APP 在较少改动的情况下达到类似效果；
这些系统控制接口包括：隐藏/显示系统状态栏、隐藏/显示底部导航栏，详见 Demo 代码样例；

5.2 修改屏幕密度

考虑到部分旧 APP 并未适配 1080P 分辨率的显示屏，因此，提供了动态修改显示密度接口，以使旧版本 APP 不需要重新做分辨率适配。但是，并不推荐这种方案，因为这是对 1080P 分辨率和更好用户体验的浪费。

保密信息

梅特勒-托利多(常州)测量技术有限公司
Mettler-Toledo (Changzhou) Measurement Technology Ltd.

地址: 江苏省常州市新北区
太湖西路111号
电话: 86-0519-86642040
传真: 86-0519-86641991
邮编: 213125
Address: 111 Taihu West Road,
Changzhou, Jiangsu
Tel: 86-0519-86642040
Fax: 86-0519-86641991
Zip: 213125
www.mt.com

详见 Demo 代码样例;

6. 返回码及含义

返回码	含义	备注
0	命令执行成功或打印机空闲状态	本返回码表示成功或可用，非错误。
-1	到 MT 称重或打印服务连接未建立	是否重复建立通讯连接; 是否误删除了 MT 的服务 APP; 其它, 请联系 MT
-2	到 MT 的远程通信异常出错	内部错误, 请联系 MT
-3	没有这个设备或接口	检查打印机类型是否正确; 设备供电是否异常; 请联系 MT
-4	无效的参数	一般是参数范围出错;
-5	接口或设备未初始化完成	调用逻辑错误或内部错误, 请联系 MT
-6	收到设备表示失败或出错的应答	内部错误, 请联系 MT
-7	通信失败	内部通信错误, 请联系 MT
-8	不支持的功能	当前的硬件类型不支持该功能
-9	命令的请求和应答不匹配	内部错误, 重启秤后如不能恢复, 请联系 MT
-10	发送消息出错	内部错误, 请联系 MT
-11	权限不够	内部错误, 请联系 MT
-12	内部数据格式错误	内部错误, 请联系 MT
-13	未知错误	可能是 JAVA 层面函数异常; 内部错误, 请联系 MT

-14	端口未准备好	读写操作时通信端口没打开, 请联系 MT
-1000	重量不稳定, 是中间过渡状态, UI 上可以显示, 但不能作为最终称重结果	秤盘晃动导致, 一般等待短暂时间会稳定; 否则, 请检查环境导致晃动原因
-1001	欠载	秤盘上重量过小, 无法称重, 检查是否没放秤盘; 可能最近一次秤开机时秤盘有物体导致零点偏差较大, 需拿掉秤盘上物体, 彻底断电 10 秒钟后重新接电开机。
-1002	过载	秤盘上重量太大, 减少秤盘上物品重量
-1003	无效状态	一般是秤初始化时没设置好零点, 可拔掉电源线, 清空秤盘上物体后, 约 10 秒后, 再重新接电开机。
-1004	加密通道出错	内部错误, 如整秤重启后不能恢复, 请联系 MT
-1005	不符合法规	根据称量法规, 在有称量皮重的情况下不能设置预置皮重, 必须先手工清除称量皮重, 再设置预置皮重; 反过来一样, 在有预置皮重的情况下, 不能设置称量皮重, 需先去掉预置皮重, 再设置称量皮重。
-2001	打印机未连接	内部异常, 请联系 MT
-2002	打印机正忙	如打印机实际不在打印数据, 且几秒后还不能恢复正常, 请联系 MT
-2003	打印机缺纸	检查是否缺纸, 如不缺则再做标定打印机缺纸传感器操作; 如还不能恢复正常, 请联系 MT
-2004	打印机过热	等待一会儿再打印, 如不能解决请联系 MT
-2005	打印的标签未取走	取走打印口的标签或关闭取纸传感器; 如不能解决请联系 MT
-2006	非法打印头	确认打印机是否梅特勒-托利多正品备件
-2007	打印头或打印舱门未关闭	检查打印头是否压紧打印纸; 检测打印机纸舱门是否关闭;
-2008	切刀出错	更换打印机或联系 MT
-2009	切刀需要清洁	清洁打印机切刀刀片
-2010	打印机需要标定打印纸	放入打印纸, 执行标定打印机指令

保密信息

梅特勒-托利多(常州)测量技术有限公司
Mettler-Toledo (Changzhou) Measurement Technology Ltd.

地址: 江苏省常州市新北区
太湖西路111号
电话: 86-0519-86642040
传真: 86-0519-86641991
邮编: 213125

Address: 111 Taihu West Road,
Changzhou, Jiangsu
Tel: 86-0519-86642040
Fax: 86-0519-86641991
Zip: 213125

www.mt.com

-2011	打印机电源异常	打印机供电电压过高或过低，联系 MT
-2012	预留打印机状态	暂无具体含意

7. 代码实例

请查看配套的 Demo 代码，或联系 MT 索取；