

标签打印机

LPAPI.js 打印接口说明

Android 版本

V1.2

DothanTech 道臻技术

2019 年 12 月 03 日

目 录

目 录.....	2
文档修改记录.....	4
1. LPAPI.js 接口文件概述	5
2. LPAPI.js 接口使用简述	5
2.1 在原生项目中导入 LPAPI.jar 包.....	5
2.2 在 WebView 控件所在的 Activity 中创建标签打印接口	5
2.3 在用户代码中调用 LPAPI.js 接口.....	6
3. 连接打印机相关方法.....	6
3.1 getAllPrinters 以字符串形式返回配对列表中所有指定型号的打印机.....	6
3.2 openPrinter 异步方式打开指定名称或型号的打印机.....	7
3.3 openPrinterSync 同步方式打开指定名称或型号的打印机.....	7
3.4 getPrinterName 获取以连接打印机名称.....	8
3.5 isPrinterOpened 判断是否已经连接打印机.....	8
3.6 cancel 取消打印任务.....	8
3.7 closePrinter 关闭当前已连接打印机.....	9
3.8 reopenPrinter 异步方式打开上次连接过的打印机.....	9
3.9 reopenPrinterSync 同步方式打开上次连接过的打印机.....	10
4. 打印任务与页面的方法说明.....	10
4.1 startJob 开始绘图任务.....	10
4.2 abortJob 取消绘图任务.....	11
4.3 endJob 结束绘图任务.....	11
4.4 commitJob 提交数据打印.....	11
4.5 startPage 开始绘制一个页面.....	12
4.6 endPage 结束一个页面的绘制.....	12
5. 打印内容参数的设置或获取.....	12
5.1 setItemOrientation 设置后续绘制内容的顺时针旋转角度.....	13
5.2 getItemOrientation 得到当前绘制内容的顺时针旋转角度.....	13
5.3 setItemHorizontalAlignment 设置后续绘制动作的水平对齐方式.....	13
5.4 getItemHorizontalAlignment 得到当前绘制动作的水平对齐方式.....	14
5.5 setItemVerticalAlignment 设置后续绘制动作的垂直对齐方式.....	14
5.6 getItemVerticalAlignment 得到当前绘制动作的垂直对齐方式.....	14
5.7 setItemPenAlignment 设置线条画笔对齐方式.....	15
5.8 getItemPenAlignment 得到线条画笔对齐方式.....	15
5.9 setDrawParam 设置绘制相关的参数值.....	15
6. 绘制文本字符串方法说明.....	16

6.1	文本绘制时统一使用的字体	16
6.2	drawText 绘制文本字符串	17
7.	绘制条码方法说明	18
7.1	draw1Dbarcode 绘制 Barcode 一维条码	18
7.2	draw2DQRCode 绘制 QrCode 二维码	19
7.3	draw2DPdf417 绘制 Pdf417 二维码	19
8.	矢量图的绘制方法说明	20
8.1	drawRectangle 以指定的线宽，绘制矩形框	20
8.2	fillRectangle 以指定的线宽，绘制矩形框	20
8.3	drawRoundRectangle 以指定的线宽，绘制圆角矩形框	21
8.4	fillRoundRectangle 以指定的线宽，绘制圆角矩形框	21
8.5	drawEllipse 以指定的线宽，绘制圆	22
8.6	fillEllipse 以指定的线宽，绘制圆	22
8.7	drawCircle 以指定的位置为中心，绘制圆	23
8.8	fillCircle 打印填充圆形	23
8.9	drawLine 绘制线（直线/斜线）	24
8.10	drawDashLine 绘制点划线	24
9.	图片绘制方法说明	25
9.1	drawImage 绘制指定文件的图片	25
9.2	drawImageWithActualSize 绘制给定图片实际大小的图片	26
9.3	drawImageWithThreshold 绘制灰度阈值的图片	26

文档修改记录

序号	版本号	修改内容	修改者	修改日期
1	V0.1	文档建立	胡殿兴	2017-09-09
2	V0.2	文档完善	杨玲美	2017-10-10
3	V1.0	文档格式完善	杨玲美	2017-11-11
4	V1.1	新增 setDrawParam 函数	杨玲美	2018-03-20
5	V1.2	修改文档格式	杨玲美	2019-12-03

1. LPAPI.js 接口文件概述

LPAPI.js 接口方法说明文档，是在原生 java 对接包基础上通过 JavaScript 进行的二次封装，方便用户在二次开发中调用接口，缩短开发周期，加快开发进度。

LPAPI.js 接口为调用者提供易用的方法完成标签绘图的操作。本接口中提供了文字、一维条码、二维码、图片和各种图形等多种对象的绘制方法，同时还能进行绘制对象的旋转和绘图页面的旋转，调用者还可调用方法获得绘制完成的标签图片用于标签预览，使标签绘图的操作更加简便。

2. LPAPI.js 接口使用简述

LPAPI 接口的基本调用流程简单如下：

2.1 在原生项目中导入 LPAPI.jar 包

2.2 在 WebView 控件所在的 Activity 中创建标签打印接口

```
private LPAPI mPrinter;
private WebView mWeb;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mWeb = (WebView) findViewById(R.id.webviewer);
    ...
    // 在WebView加载html文件之前初始化标签打印接口
    mPrinter = LPAPI.Factory.createInstance(mWeb);
    // 加载html文件
    mWeb.loadUrl("file:///android_asset/index.html");
}
```

2.3 在用户代码中调用 LPAPI.js 接口

在需要调用的地方引入 LPAPI.js 文件，并在需要打印的地方调用 LPAPI.js 相关方法进行打印，参考代码如下：

```
// 得到 LPAPI 对象
var api = new LPAPI ();
// 连接配对的第一个打印机
api.openPrinter("");
// 开始绘图任务，传入参数(页面宽度，页面高度)
api.startJob(40, 30, 0);
// 开始一个页面的绘制，绘制文本字符串
api.drawText(content, 4, 5, 40, 30, 4, 0);
// 结束绘图任务提交打印
api.commitJob();
```

3. 连接打印机相关方法

3.1 getAllPrinters 以字符串形式返回配对列表中所有指定型号的打印机

功能简介	以字符串形式获取所有指定型号的已配对打印机名称。	
方法原型	LPAPI.prototype.getAllPrinters = function (printerName);	
参数名称	printerName	打印机型号过滤器，可以为打印机型号（如：DT20S）或打印机名称（DT20S-71000001）； 可以指定多个打印机型号或打印机名称，之间用英文逗号“,”进行分隔，如：“DT20S,DP20S”； 如果未指定则返回所有已配对打印机名称。
返回值	打印机名称列表字符串，多个打印机名称之间以英文逗号“,”分隔。	
使用说明	打印机名称列表需要用户自己根据英文逗号“,”进行分隔。	
使用注意		

3.2 openPrinter 异步方式打开指定名称或型号的打印机

功能简介	异步方式打开指定名称或类型的打印机。	
方法原型	LPAPI.prototype.openPrinter = function (printerName);	
参数名称	printerName	参数可为下列类型的数据： 1、 打印机屏幕上显示的名称：用于打开指定名称的打印机； 2、 打印机型号(名称前缀) :打开配对列表中的第一台对应型号的打印机； null 或者 "" ，打开配对列表中的第一台打印机；
返回值	成功与否	
使用说明	连接打印机。	
使用注意	在开始打印任务前，须先调用该方法，或者同系列的方法。	

3.3 openPrinterSync 同步方式打开指定名称或型号的打印机

功能简介	同步方式打开指定名称或型号的打印机。	
方法原型	LPAPI.prototype.openPrinterSync = function (printerName);	
参数名称	printerName	参数可为下列类型的数据： 3、 打印机屏幕上显示的名称：用于打开指定名称的打印机； 4、 打印机型号(名称前缀) :打开配对列表中的第一台对应型号的打印机； 5、 null 或者 "" ，打开配对列表中的第一台打印机；
返回值	成功与否	
使用说明		
使用注意	确保打印机已经打开，并进行了配对操作。	

3.4 getPrinterName 获取以连接打印机名称

功能简介	获取已连接打印机名称。
方法原型	LPAPI.prototype.getPrinterName = function ();
参数名称	无
返回值	如果已连接打印机，则返回对应打印机名称，否则返回 null。
使用说明	
使用注意	

3.5 isPrinterOpened 判断是否已经连接打印机

功能简介	判断是否已经连接打印机。
方法原型	LPAPI.prototype.isPrinterOpened = function ();
参数名称	无
返回值	true：表示打印机以连接成功，可以进行打印；false：表示未连接打印机，在打印之前需要连接打印机。
使用说明	
使用注意	

3.6 cancel 取消打印任务

功能简介	取消打印任务。
方法原型	LPAPI.prototype.cancel = function ();
参数名称	无

返回值	无
使用说明	取消已经提交（执行过 commitJob 函数），但尚未开始的打印任务；
使用注意	

3.7 closePrinter 关闭当前已连接打印机

功能简介	关闭已连接打印机。
方法原型	LPAPI.prototype.closePrinter = function ();
参数名称	无
返回值	无
使用说明	如果需要多个用户公用同一台打印机，在打印完成之后，可以通过该函数来断开打印机，否则其他用户无法连接到该打印机；
使用注意	1、在打印未完成之前尽量避免调用该函数，否则有可能导致打印任务终止；

3.8 reopenPrinter 异步方式打开上次连接过的打印机

功能简介	异步方式打开上次连接过的打印机。
方法原型	LPAPI.prototype.reopenPrinter = function ();
参数名称	无
返回值	成功与否
使用说明	
使用注意	返回成功仅仅表示操作被提交，并不代表着连接成功了，具体的连接结果会通过回调函数给出通知。

3.9 reopenPrinterSync 同步方式打开上次连接过的打印机

功能简介	同步方式打开上次连接过的打印机。
方法原型	LPAPI.prototype.reopenPrinterSync = function ();
参数名称	无
返回值	成功与否
使用说明	
使用注意	

4. 打印任务与页面的方法说明

4.1 startJob 开始绘图任务

功能简介	开始绘图任务。	
方法原型	LPAPI.prototype.startJob = function (width, height, orientation);	
参数名称	width	标签宽度（基于绘制视图，不考虑标签旋转。单位毫米）。
	height	标签高度（基于绘制视图，不考虑标签旋转。单位毫米）。
	orientation	绘图任务的标签页面顺时针旋转角度，0/90/180/270，默认值 0。
返回值	成功与否	
使用说明	开始一个绘图任务时需调用此方法。	
使用注意	开始绘图任务时，之前的任务/数据将会被全部丢弃。	

4.2 abortJob 取消绘图任务

功能简介	取消绘图任务。
方法原型	LPAPI.prototype.abortJob = function ();
参数名称	无
返回值	成功与否
使用注意	当前的任务/数据将会被全部丢弃，但是参数设置将会被保留。

4.3 endJob 结束绘图任务

功能简介	结束绘图任务。
方法原型	LPAPI.prototype.endJob = function ();
参数名称	无
返回值	无

4.4 commitJob 提交数据打印

功能简介	提交数据打印。
方法原型	LPAPI.prototype.commitJob = function ();
参数名称	无
返回值	成功与否

4.5 startPage 开始绘制一个页面

功能简介	开始绘制一个页面。
方法原型	LPAPI.prototype.startPage = function ();
参数名称	无
返回值	成功与否

4.6 endPage 结束一个页面的绘制

功能简介	结束一个页面的绘制。
方法原型	LPAPI.prototype.endPage = function ();
参数名称	无
返回值	无

5. 打印内容参数的设置或获取

[ItemAlignment](#) 绘制条码的几种风格：

数值	描述
0	水平靠左/垂直靠上
1	水平居中/垂直居中
2	水平靠右/垂直靠下
3	对象子元素的对齐方式同对象的对齐方式，当前在一维码文本的水平对齐方式中使用

[PenAlignment](#) 绘制的字体有如下几种风格：

数值	描述
0	绘制的线以指定的位置为中央

1	绘制的线在指定的位置内侧
---	--------------

5.1 setItemOrientation 设置后续绘制内容的顺时针旋转角度

功能简介	设置后续绘制内容的顺时针旋转角度。	
方法原型	LPAPI.prototype.setItemOrientation = function (orientation);	
参数名称	orientation	后续绘制内容的顺时针旋转角度（0/90/180/270），默认值为0。
返回值	无	

5.2 getItemOrientation 得到当前绘制内容的顺时针旋转角度

功能简介	得到当前绘制内容的顺时针旋转角度。	
方法原型	LPAPI.prototype.getItemOrientation = function ();	
参数名称	无	
返回值	当前绘制内容的顺时针旋转角度（0/90/180/270）	

5.3 setItemHorizontalAlignment 设置后续绘制动作的水平对齐方式

功能简介	设置后续绘制动作的水平对齐方式。	
方法原型	LPAPI.prototype.setItemHorizontalAlignment = function (alignment);	
参数名称	alignment	后续绘制动作的水平对齐方式（ ItemAlignment ），默认值为 ItemAlignment.LEFT 。
返回值	无	
使用说明	水平方向是基于后续绘制的内容的水平绘制方向，跟随绘制内容的顺时针旋转角度旋转。	

5.4 getItemHorizontalAlignment 得到当前绘制动作的水平对齐方式

功能简介	得到当前绘制动作的水平对齐方式。
方法原型	LPAPI.prototype.getItemHorizontalAlignment = function ();
参数名称	无
返回值	当前绘制动作的水平对齐方式 (ItemAlignment)
使用说明	水平方向是基于后续绘制的内容的水平绘制方向，跟随绘制内容的顺时针旋转角度旋转。

5.5 setItemVerticalAlignment 设置后续绘制动作的垂直对齐方式

功能简介	设置后续绘制动作的垂直对齐方式。
方法原型	LPAPI.prototype.setItemVerticalAlignment = function (int alignment);
参数名称	alignment 后续绘制动作的垂直对齐方式 (ItemAlignment)，默认值为 ItemAlignment.TOP 。
返回值	无
使用说明	垂直方向是基于后续绘制的内容的垂直绘制方向，跟随绘制内容的顺时针旋转角度旋转。

5.6 getItemVerticalAlignment 得到当前绘制动作的垂直对齐方式

功能简介	得到当前绘制动作的垂直对齐方式。
方法原型	LPAPI.prototype.getItemVerticalAlignment = function ();
参数名称	无
返回值	当前绘制动作的垂直对齐方式 (ItemAlignment)
使用说明	垂直方向是基于后续绘制的内容的垂直绘制方向，跟随绘制内容的顺时针旋转角度旋转。

5.7 setItemPenAlignment 设置线条画笔对齐方式

功能简介	设置线条画笔对齐方式。	
方法原型	LPAPI.prototype.setItemPenAlignment = function (penAlignment);	
参数名称	penAlignment	线条画笔对齐方式（ PenAlignment ），数值为以下两者之一： PenAlignment.CENTER ：绘制的线以指定的位置为中央； PenAlignment.INSET ：绘制的线在指定的位置内侧。
返回值	无	

5.8 getItemPenAlignment 得到线条画笔对齐方式

功能简介	得到线条画笔对齐方式。	
方法原型	LPAPI.prototype.getItemPenAlignment = function ();	
参数名称	无	
返回值	线条画笔对齐方式（ PenAlignment ），数值为以下两者之一： PenAlignment.CENTER ：绘制的线以指定的位置为中央； PenAlignment.INSET ：绘制的线在指定的位置内侧。	

5.9 setDrawParam 设置绘制相关的参数值

功能简介	设置绘制相关的参数值。	
方法原型	LPAPI.prototype.setDrawParam = function (name, value);	
参数名称	name	参数名称，在 DrawParamName 接口的静态字符串变量中定义。 FONT_NAME ：存放在工程 assets 目录中的字体文件名称，值为 String 类型。如：HeiTi.ttf。注意是字体文件名称，而不一定是字体名称。参数值没有指定后缀的情况下，会自动加上 .ttf 后缀。 QR_CODE_VERSION ：QRCode 的编码版本号（1~40），值为 Integer 类型。具体含义详见 QRCode 编码规范。未指定版本号时表示根据编码内容自动采用适合的编码版本号。

		<p>ERROR_CORRECTION : QRCode 的纠错级别 (0 ~ 3) , 值为 Integer 类型。默认为 0。对于标签打印, 建议采用纠错级别 0, 以便打印黑点尽量大, 便于二维码更容易扫描识别。</p> <p>CHARACTER_SET : QRCode 的字符串编码类型, 值为 String 类型。默认为 UTF-8。</p> <p>MARGIN : QRCode 二维码的四边留白, 值为 Integer 类型。QRCode 编码规范中定义的留白值为 0/2/4。默认是 0, 也即不留白。</p>
	value	参数值, 具体含义参考不同的参数名称。
返回值	无	

6. 绘制文本字符串方法说明

[FontStyle](#) 绘制的字体有如下几种风格：

数值	描述
0	正常
1	粗体
2	斜体
3	粗斜体
4	下划线
8	删除线

6.1 文本绘制时统一使用的字体

安卓系统中不同品牌型号的手机的默认字体, 有很大的差异, 因此对于打印出的标签的一致性要求比较高的系统, 有必要对绘制文本时使用的字体进行统一。统一文本的字体可以通过下面两种方法来进行：

- 对于整个标签中使用一种字体的情况, 可以直接在安卓 APP 工程的 assets 目录下, 放入要被使用的这个字体文件, 将文件名称修改为 FONT.ttf。绘制文本的接口函数就会使用该字体, 进行文本的绘制。在找不到该字体文件的情况下, 文本绘制函数才会使用系统默认的字体；

- 对于整个标签中使用多种字体的情况，可以在安卓 APP 工程的 assets 目录下，放入需要使用的多个字体文件（对于中文名称的字体文件，建议将文件名称改成英文，因为有的安卓系统对 assets 目录下的中文名称文件，支持的不太好），然后在绘制文本之前，通过调用 [setDrawParam\(IAtBitmap.DrawParamName.FONT_NAME, "xxx" \)](#) 来指定字体文件，进行文本的绘制。

6.2 drawText 绘制文本字符串

功能简介	绘制文本字符串。	
方法原型	LPAPI.prototype.drawText = function (text, x, y, width, height, fontHeight, fontStyle);	
参数名称	text	需要绘制的文本字符串。
	x	绘制的文本框左上角水平位置（单位毫米）。
	y	绘制的文本框左上角垂直位置（单位毫米）。
	width	绘制的文本框水平宽度（单位毫米）。如果 width 为 0，则会根据绘制文本的显示宽度，根据当前对齐方式进行以 x 为基准点的左中右对齐。 默认值为 0。
	height	绘制的文本框垂直高度（单位毫米）。如果 height 为 0，则会根据绘制文本的显示高度，根据当前对齐方式进行以 y 为基准点的上中下对齐。 默认值为 0。
	fontHeight	文字大小（单位毫米）。
	fontStyle	字体风格，正常、粗体、斜体、粗斜体、下划线、删除线（ FontStyle ）。默认值为正常字体。
返回值	成功与否	
使用说明	<ul style="list-style-type: none">如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行文本字符串的绘制。绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。fontHeight 是以毫米为单位。 字号和毫米的转换公式为： 字号 = 毫米 * 72 / 25.4 比方说 9 号字为 3.175 毫米，12 号字为 4.233 毫米	

7. 绘制条码方法说明

[BarcodeType](#) 绘制条码的几种风格：

数值	描述
20	UPC_A
21	UPC_E
22	EAN13
23	EAN8
24	CODE39
25	ITF25
26	CODEBAR
27	CODE93
28	CODE128
29	ISBN
30	ECODE39
60	AUTO

7.1 draw1Dbarcode 绘制 Barcode 一维条码

功能简介	绘制一维条码。	
方法原型	LPAPI.prototype.draw1Dbarcode = function (text, type, x, y, width, height, textHeight);	
参数名称	text	需要绘制的一维条码的内容。
	type	一维条码的编码类型 (BarcodeType) , 默认值为 60。
	x	绘制的一维条码的左上角水平位置 (单位毫米) 。
	y	绘制的一维条码的左上角垂直位置 (单位毫米) 。
	width	一维条码的整体显示宽度。
	height	一维条码的显示高度 (包括供人识读文本) 。
	textHeight	供人识读文本的高度 (单位毫米) , 建议为 3 毫米。
返回值	成功与否	

使用说明	<ul style="list-style-type: none">如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行一维条码的绘制。绘制的左上角水平垂直位置、宽度、高度是基于当前绘制页面的水平垂直方向。
------	---

7.2 draw2DQRCode 绘制 QrCode 二维码

功能简介	绘制 QrCode 二维码。	
方法原型	LPAPI.prototype.draw2DQRCode = function (text, x, y, width);	
参数名称	text	需要绘制的 QrCode 二维码的内容。
	x	绘制的 QrCode 二维码的左上角水平位置（单位毫米）。
	y	绘制的 QrCode 二维码的左上角垂直位置（单位毫米）。
	width	绘制的 QrCode 二维码的水平宽度（单位毫米）。
返回值	成功与否	
使用说明	<ul style="list-style-type: none">如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行 QrCode 二维码的绘制。绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。	

7.3 draw2DPdf417 绘制 Pdf417 二维码

功能简介	绘制 Pdf417 二维码。	
方法原型	LPAPI.prototype.draw2DPdf417 = function (text, x, y, width, height);	
参数名称	text	需要绘制的 Pdf417 二维码的内容。
	x	绘制的 Pdf417 二维码的左上角水平位置（单位毫米）。
	y	绘制的 Pdf417 二维码的左上角垂直位置（单位毫米）。
	width	绘制的 Pdf417 二维码的水平宽度（单位毫米）。
	height	绘制的 Pdf417 二维码的垂直高度（单位毫米）。
返回值	成功与否	

使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法,则本方法会自动调用 startPage 开始绘制一个页面, 然后进行 Pdf417 二维码的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。
------	--

8. 矢量图的绘制方法说明

8.1 drawRectangle 以指定的线宽, 绘制矩形框

功能简介	以指定的线宽, 绘制矩形框。	
方法原型	LPAPI.prototype.drawRectangle = function (x, y, width, height, lineWidth);	
参数名称	x	绘制的矩形框的左上角水平位置 (单位 mm)。
	y	绘制的矩形框的左上角垂直位置 (单位 mm)。
	width	绘制的矩形框的水平宽度 (单位 mm)。
	height	绘制的矩形框的垂直高度 (单位 mm)。
	lineWidth	矩形框的线宽 (单位 mm)。矩形框的线宽是向矩形框内部延伸的。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法,则本方法会自动调用 startPage 开始绘制一个页面, 然后进行矩形框的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

8.2 fillRectangle 以指定的线宽, 绘制矩形框

功能简介	以指定的线宽, 绘制矩形框。	
方法原型	LPAPI.prototype.fillRectangle = function (x, y, width, height);	
参数名称	x	绘制的填充矩形框的左上角水平位置 (单位 mm)。
	y	绘制的填充矩形框的左上角垂直位置 (单位 mm)。
	width	绘制的填充矩形框的水平宽度 (单位 mm)。
	height	绘制的填充矩形框的垂直高度 (单位 mm)。

返回值	成功与否
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法,则本方法会自动调用 startPage 开始绘制一个页面, 然后进行矩形框的绘制。 <p>绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。</p>

8.3 drawRoundRectangle 以指定的线宽, 绘制圆角矩形框

功能简介	以指定的线宽, 绘制圆角矩形框。	
方法原型	LPAPI.prototype.drawRoundRectangle = function (x, y, width, height, cornerWidth, cornerHeight, lineWidth);	
参数名称	x	绘制的圆角矩形框的左上角水平位置 (单位 mm)。
	y	绘制的圆角矩形框的左上角垂直位置 (单位 mm)。
	width	绘制的圆角矩形框的水平宽度 (单位 mm)。
	height	绘制的圆角矩形框的垂直高度 (单位 mm)。
	cornerWidth	圆角宽度 (单位 mm)。
	cornerHeight	圆角高度 (单位 mm), 默认值为同 cornerWidth。
	lineWidth	圆角矩形框的线宽 (单位 mm)。圆角矩形框的线宽是向圆角矩形框内部延伸的。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法,则本方法会自动调用 startPage 开始绘制一个页面, 然后进行圆角矩形框的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

8.4 fillRoundRectangle 以指定的线宽, 绘制圆角矩形框

功能简介	以指定的线宽, 绘制圆角矩形框。	
方法原型	LPAPI.prototype.drawRoundRectangle = function (x, y, width, height, cornerWidth, cornerHeight);	
参数名称	x	绘制的填充圆角矩形框的左上角水平位置 (单位 mm)。
	y	绘制的填充圆角矩形框的左上角垂直位置 (单位 mm)。

	width	绘制的填充圆角矩形框的水平宽度（单位 mm）。
	height	绘制的填充圆角矩形框的垂直高度（单位 mm）。
	cornerWidth	圆角宽度（单位 mm）。
	cornerHeight	圆角高度（单位 mm），默认值为同 cornerWidth。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行圆角矩形框的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。	

8.5 drawEllipse 以指定的线宽，绘制圆

功能简介	以指定的线宽，绘制圆。	
方法原型	LPAPI.prototype.drawEllipse = function (x, y, width, height, lineWidth);	
参数名称	x	绘制的椭圆的左上角水平位置（单位 mm）。
	y	绘制的椭圆的左上角垂直位置（单位 mm）。
	width	绘制的椭圆的水平宽度（单位 mm）。
	height	绘制的椭圆的垂直高度（单位 mm），默认值为同 width，也即绘制圆。
	lineWidth	椭圆的线宽（单位 mm）。椭圆的线宽是向椭圆内部延伸的。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行椭圆/圆的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 width 和 height 相等时就是打印圆。 	

8.6 fillEllipse 以指定的线宽，绘制圆

功能简介	以指定的线宽，绘制圆。
------	-------------

方法原型	LPAPI.prototype.drawEllipse = function (x, y, width, height);	
参数名称	x	绘制的填充椭圆的左上角水平位置（单位 mm）。
	y	绘制的填充椭圆的左上角垂直位置（单位 mm）。
	width	绘制的填充椭圆的水平宽度（单位 mm）。
	height	绘制的填充椭圆的垂直高度（单位 mm），默认值为同 width，也即绘制圆。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行椭圆/圆的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。width 和 height 相等时就是打印圆。 	

8.7 drawCircle 以指定的位置为中心，绘制圆

功能简介	以指定的线宽，绘制椭圆。	
方法原型	LPAPI.prototype.drawEllipse = function (x, y, radius, lineWidth);	
参数名称	x	绘制的圆的 x 轴中心位置（单位 mm）。
	y	绘制的圆的 y 轴中心位置（单位 mm）。
	radius	绘制圆的半径（单位 mm）。
	lineWidth	椭圆的线宽（单位 mm）。椭圆的线宽是向椭圆内部延伸的。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行椭圆/圆的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。width 和 height 相等时就是打印圆。 	

8.8 fillCircle 打印填充圆形

功能简介	以指定的线宽，绘制圆。
------	-------------

方法原型	LPAPI.prototype.fillCircle = function (x, y, radius);	
参数名称	x	绘制的填充圆形 X 轴坐标中心点位置 (单位 mm)。
	y	绘制的填充圆形 y 轴坐标中心点位置 (单位 mm)。
	radius	绘制填充圆形的半径 (单位 mm)。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> ● 如果之前没有调用 startPage 而直接调用本方法,则本方法会自动调用 startPage 开始绘制一个页面, 然后进行椭圆/圆的绘制。 ● 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 ● width 和 height 相等时就是打印圆。 	

8.9 drawLine 绘制线 (直线/斜线)

功能简介	绘制线 (直线/斜线)。	
方法原型	LPAPI.prototype.drawLine = function (x1, y1, x2, y2, lineWidth);	
参数名称	x1	线的起点的水平位置 (单位 mm)。
	y1	线的起点的垂直位置 (单位 mm)。
	x2	线的终点的水平位置 (单位 mm)。
	y2	线的终点的垂直位置 (单位 mm)。
	lineWidth	线宽 (单位 mm)。线宽是向线的下方延伸的。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> ● 如果之前没有调用 startPage 而直接调用本方法,则本方法会自动调用 startPage 开始绘制一个页面, 然后进行直线/斜线的绘制。 ● 绘制的水平垂直位置是基于当前绘制页面的水平垂直方向。 	

8.10 drawDashLine 绘制点划线

功能简介	绘制点划线。
------	--------

方法原型	LPAPI.prototype.drawDashLine = function (x1, y1, x2, y2, lineWidth, dashLen); LPAPI.prototype.drawDashLine2 = function (x1, y1, x2, y2, lineWidth, dashLen1, dashLen2); LPAPI.prototype.drawDashLine4 = function (x1, y1, x2, y2, lineWidth, dashLen1, dashLen2, dashLen3, dashLen4);	
参数名称	x1	线的起点的水平位置（单位 mm）。
	y1	线的起点的垂直位置（单位 mm）。
	x2	线的终点的水平位置（单位 mm）。
	y2	线的终点的垂直位置（单位 mm）。
	lineWidth	线宽（单位 mm）。线宽是向线的下方延伸的。
	dashLen1	点划线第一段的长度（单位 mm）。
	dashLen2	点划线第二段的长度（单位 mm）。
	dashLen3	点划线第三段的长度（单位 mm）。
	dashLen4	点划线第四段的长度（单位 mm）。
	dashLen	点划线线段长度的数组（单位 mm）。
	dashCount	点划线线段长度数组的元素个数。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行点划线的绘制。 绘制的水平垂直位置是基于当前绘制页面的水平垂直方向。 	

9. 图片绘制方法说明

9.1 drawImage 绘制指定文件的图片

功能简介	绘制指定文件的图片。	
方法原型	LPAPI.prototype.drawImage = function (imageFile, x, y, width, height);	
参数名称	imageFile	图片路径。

	x	打印对象在水平方向上的位置（单位 mm）。
	y	打印对象在垂直方向上的位置（单位 mm）。
	width	打印对象的宽度（单位 mm）。如果 width 为 0，则采用加载的位图的宽度。默认值为 0。
	height	打印对象的高度（单位 mm）。如果 height 为 0，则采用加载的位图的高度。默认值为 0。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行图片的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

9.2 drawImageWithActualSize 绘制给定图片实际大小的图片

功能简介	绘制给定图片实际大小的图片。	
方法原型	LPAPI.prototype.drawImageWithActualSize = function (imageFile, x, y);	
参数名称	imageFile	图片路径。
	x	打印对象在水平方向上的位置（单位 mm）。
	y	打印对象在垂直方向上的位置（单位 mm）。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行图片的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

9.3 drawImageWithThreshold 绘制灰度阈值的图片

功能简介	绘制指定文件的图片。	
方法原型	LPAPI.prototype.drawImageWithThreshold = function (imageFile, x, y, width, height, threshold);	
参数名称	imageFile	图片路径。
	x	打印对象在水平方向上的位置（单位 mm）。

	y	打印对象在垂直方向上的位置（单位 mm）。
	width	打印对象的宽度（单位 mm）。如果 width 为 0，则采用加载的位图的宽度。默认值为 0。
	height	打印对象的高度（单位 mm）。如果 height 为 0，则采用加载的位图的高度。默认值为 0。
	threshold	绘制位图的灰度阈值。256 表示绘制灰度图片，257 表示绘制原色图片，0 ~ 255 表示绘制黑白图片，原图颜色 > 灰度阈值的点会被认为是白色，而原图颜色 <= 灰度阈值的点会被认为是黑色。
返回值	成功与否	
使用说明	<ul style="list-style-type: none">● 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行图片的绘制。● 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。	