

IOS SDK-内容排版

一、设置标签大小及视图方向

由标签视图转换为编辑视图的旋转角度方向为逆时针。接口参数中的宽高 取编辑视图的宽高

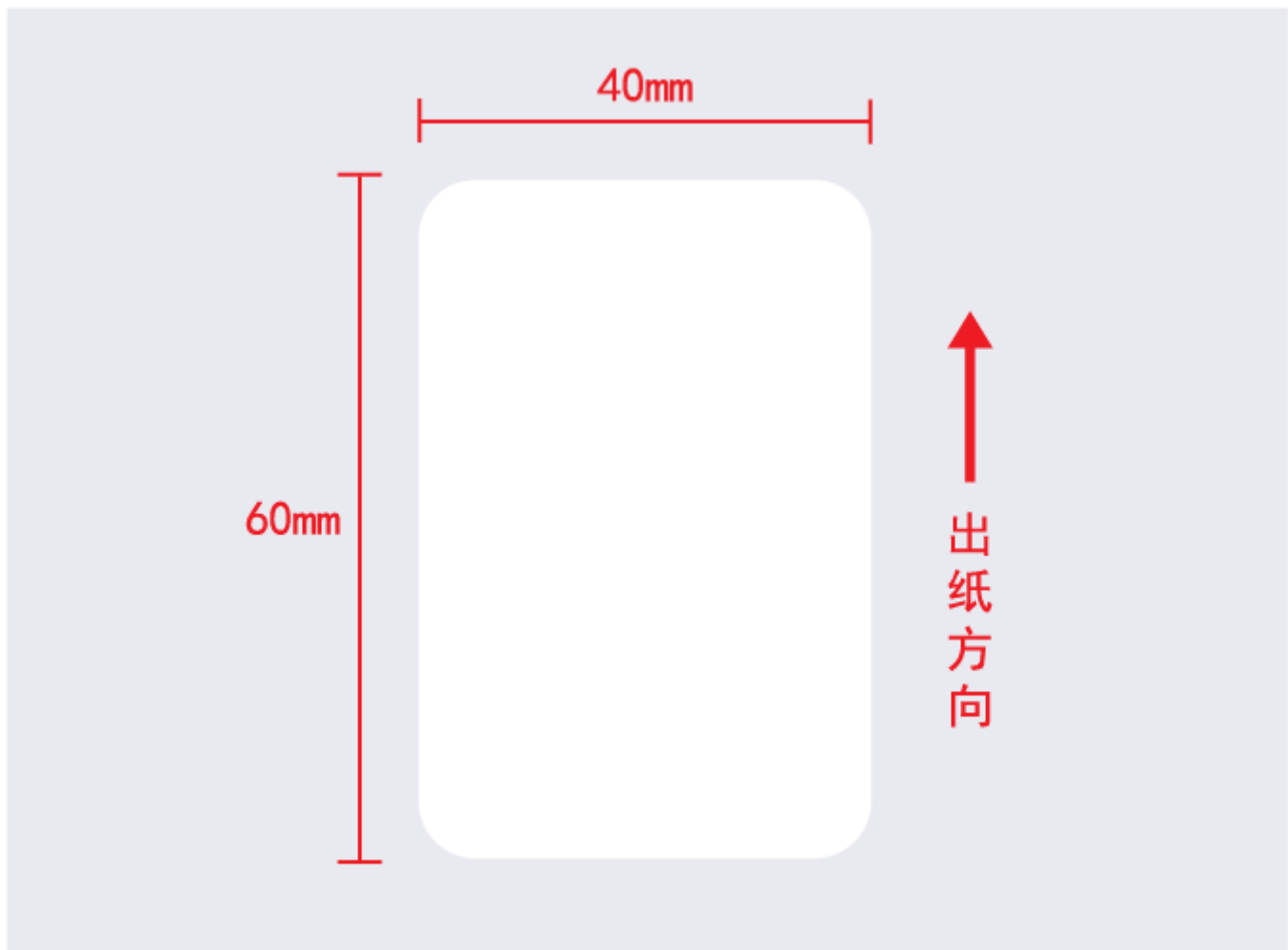
1.1 旋转0°为例

以尺寸为40-60的标签为例（单位mm）

代码块

```
1  /// 绘制画板
2  /// @param width 宽
3  /// @param height 高
4  /// @param horizontalShift 水平偏移,暂不可用
5  /// @param verticalShift 竖直偏移,暂不可用
6  /// @param rotate 旋转角度
7  /// @param font 使用字体路径,缺省输入nil
8  [JCAPI initDrawingBoard:40 withHeight:60 withHorizontalShift:0
  withVerticalShift:0 rotate:0 font:@""];
```

编辑视图示例图:



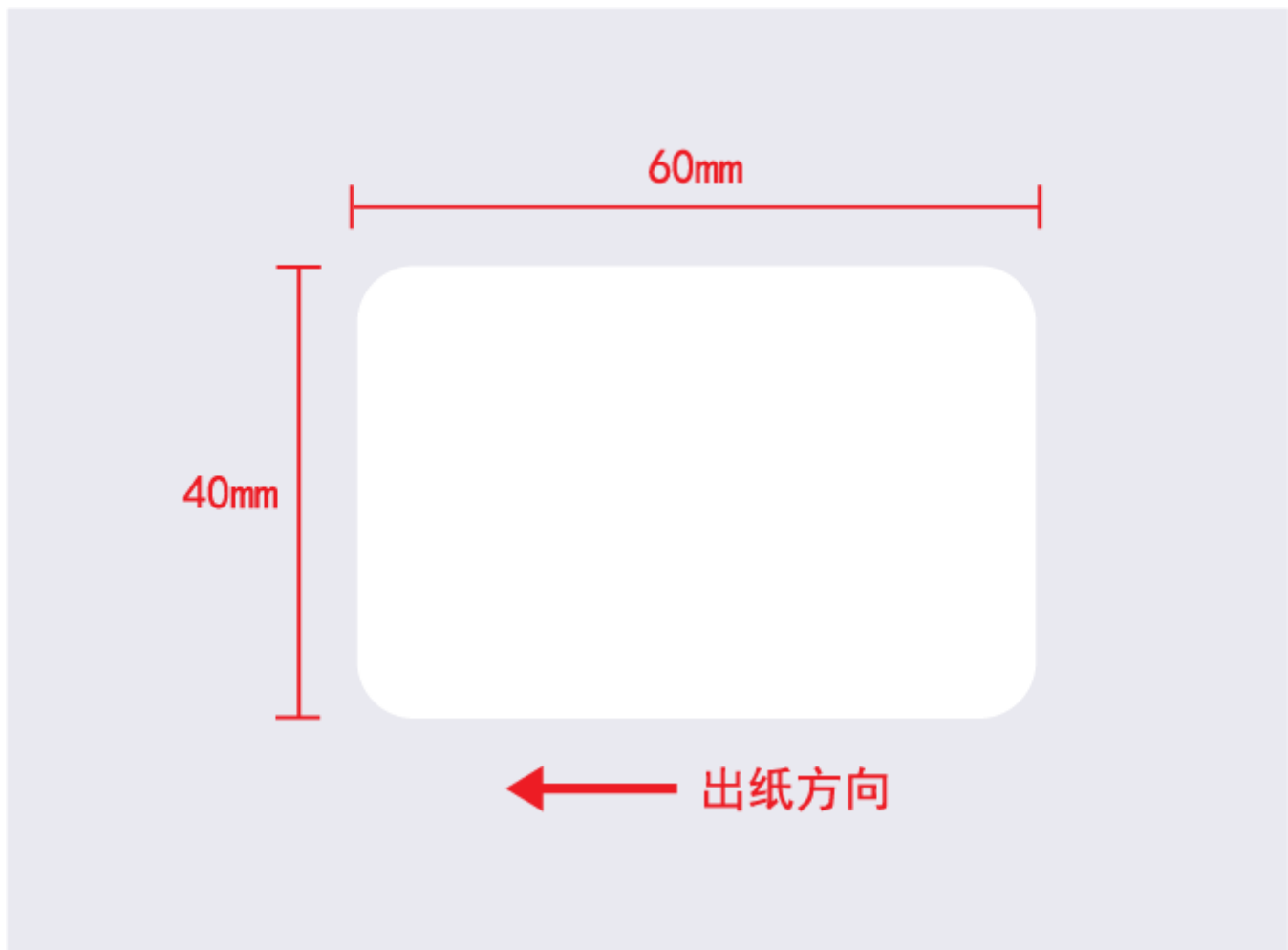
1.2 旋转90°为例

以尺寸为40-60的标签为例（单位mm）

代码块

```
1  /// 绘制画板
2  /// @param width 宽
3  /// @param height 高
4  /// @param horizontalShift 水平偏移,暂不可用
5  /// @param verticalShift 竖直偏移,暂不可用
6  /// @param rotate 旋转角度
7  /// @param font 使用字体路径,缺省输入nil
8  [JCAPI initDrawingBoard:60 withHeight:40 withHorizontalShift:0
  withVerticalShift:0 rotate:90 font:@""];
```

编辑视图示例图:



二、元素绘制

元素的绘制，基于当前编辑视图绘制。编辑视图的左上角坐标为 0,0。对齐参数或接口仅对文本框有效，是文本相对于文本框的对齐

2.1 文本框绘制

代码块

```
1  /// 绘制画板
2  /// @param width 宽
3  /// @param height 高
4  /// @param horizontalShift 水平偏移,暂不可用
5  /// @param verticalShift 竖直偏移,暂不可用
6  /// @param rotate 旋转角度
7  /// @param font 使用字体路径,缺省输入nil
8  [JCAPI initDrawingBoard:60 withHeight:40 withHorizontalShift:0
  withVerticalShift:0 rotate:90 font:@""];
9  /// 绘制文本
10 /// @param x 水平起点
11 /// @param y 竖直起点
12 /// @param w 宽
```

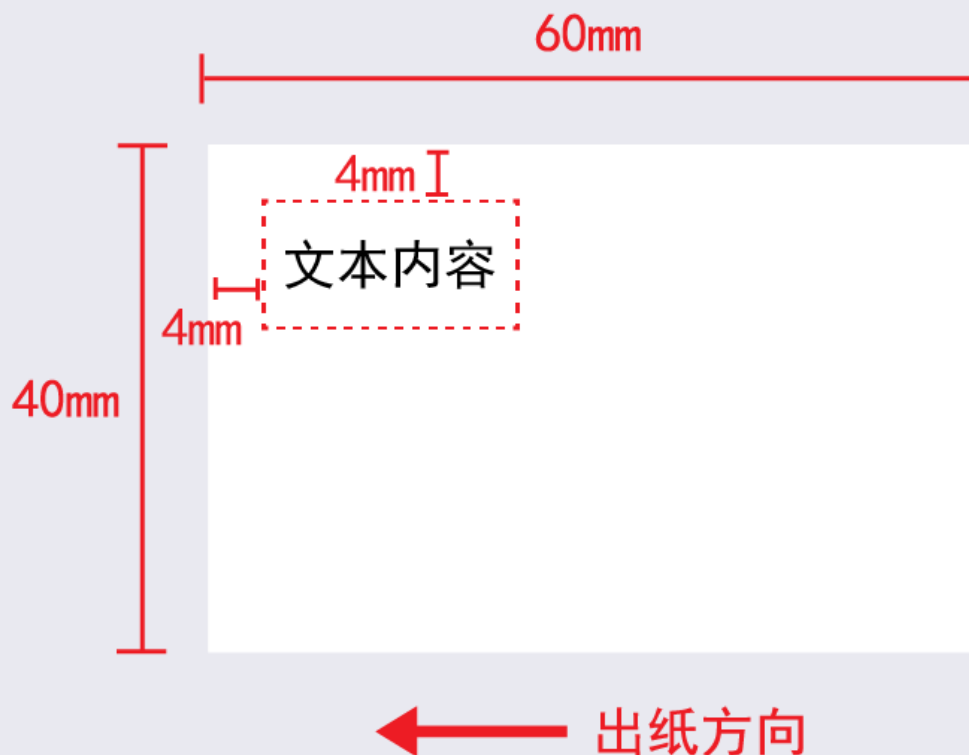
```

13  /// @param h 高
14  /// @param text 内容
15  /// @param fontFamily 字体
16  /// @param fontSize 字体大小
17  /// @param rotate 旋转
18  /// @param textAlignHorizontal 文本水平对齐方式
19  /// @param textAlignVertical 文本竖直对齐方式
20  /// @param lineMode 换行方式 1:宽高固定, 内容大小自适应 (字号/字符间距/行间距 按比例缩放) 2:宽度固定, 高度自适应 3:宽高固定, 超出内容后面加... 4:宽高固定, 超出内容直裁切 6:宽高固定, 内容超过预设宽高时自动缩小 (字号/字符间距/行间距 按比例缩放)
21  /// @param letterSpacing 字体间隔
22  /// @param lineSpacing 行间隔
23  /// @param fontStyles [0,0,0,0] 斜体,加粗,下划线,删除下划线(暂不可用)
24  [JCAPI drawLabelText:4.0 withY:4.0 withWidth: 20.0 withHeight:10.0
  withString:@"文本内容" withFontFamily:@"宋体" withFontSize:3.0 withRotate:0
  withTextAlignHorizontal:1 withTextAlignVertical:1 withLineMode:6
  withLetterSpacing:0 withLineSpacing:1 withFontStyle: @[0,0,0,0]];

```

DrawLabelText (drawText) 前五位参数分别为:文本框相对于编辑视图的X轴坐标, 文本框相对于编辑视图的Y轴坐标, 文本框的宽度, 文本框的高度, 文本框中文本的内容

效果图如下 (文本框真实绘制效果不包含边框线) :



2.2 线条绘制

代码块

```
1  /// 绘制画板
2  /// @param width 宽
3  /// @param height 高
4  /// @param horizontalShift 水平偏移,暂不可用
5  /// @param verticalShift 竖直偏移,暂不可用
6  /// @param rotate 旋转角度
7  /// @param font 使用字体路径,缺省输入nil
8  [JCAPI initDrawingBoard:60 withHeight:40 withHorizontalShift:0
withVerticalShift:0 rotate:90 font:@""];
9  /// 绘制线条
10 /// @param x 水平坐标
11 /// @param y 垂直坐标
12 /// @param w 标贴宽度,单位mm
13 /// @param h 标贴高度,单位mm
14 /// @param rotate 旋转角度,仅支持0,90,180,270
15 /// @param lineType 线条类型 1:实线 2:虚线类型,虚实比例1:1
16 /// @param dashWidth 线条为虚线宽度,【实线段长度,空线段长度】
```

```
17 [JCAPI DrawLableLine:4.0 withY:4.0 withWidth:52.0 withHeight:0.5 withRotate:0  
withLineType:1 withDashWidth:[dic valueForKey:@[1,2]]];
```

drawLabelLine参数分别为:线条起点相对于编辑视图的X轴坐标, 线条起点相对于编辑视图的Y轴坐, 线条宽度, 线条高度, 线条旋转角度, 线条类型, 虚线宽度 (虚线类型时生效) (小程序接口与其他端绘制接口存在差异)

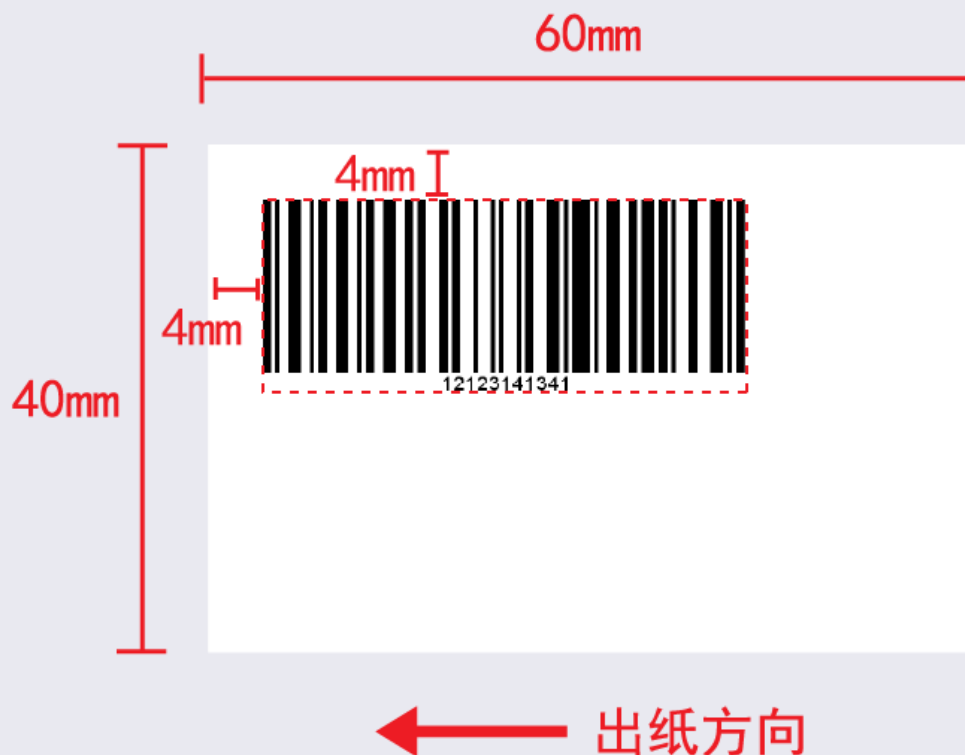
效果图如下:

2.3 一维码 (条形码) 绘制

代码块

```
1  /// 绘制画板  
2  /// @param width 宽  
3  /// @param height 高  
4  /// @param horizontalShift 水平偏移, 暂不可用  
5  /// @param verticalShift 竖直偏移, 暂不可用  
6  /// @param rotate 旋转角度  
7  /// @param font 使用字体路径, 缺省输入nil  
8  [JCAPI initDrawingBoard:60 withHeight:40 withHorizontalShift:0  
withVerticalShift:0 rotate:90 font:@""];  
9  /// 绘制条码  
10 /// @param x 水平坐标  
11 /// @param y 垂直坐标  
12 /// @param w 标贴宽度, 单位mm  
13 /// @param h 标贴高度, 单位mm  
14 /// @param text 文本内容  
15 /// @param fontSize 文本字号  
16 /// @param rotate 旋转角度, 仅支持0, 90, 180, 270  
17 /// @param codeType 一维码类型 20:CODE128 21:UPC-A 22:UPC-E 23:EAN8 24:EAN13  
25:CODE93 26:CODE39 27:CODEBAR 28:ITF25  
18 /// @param textHeight 文本高度  
19 /// @param textPosition 文本位置, int 一维码文字识别码显示位置 0:下方显示 1:上方显示  
20 2:不显示  
20 [JCAPI drawLableBarCode:4.0 withY:4.0 withWidth:38.0 withHeight:15.0  
withString:@"12123141341" withFontSize:2.0 withRotate:0 withCodeType:20  
withTextHeight:2.5 withTextPosition:0];
```

效果图如下 (文本框真实绘制效果不包含边框线):



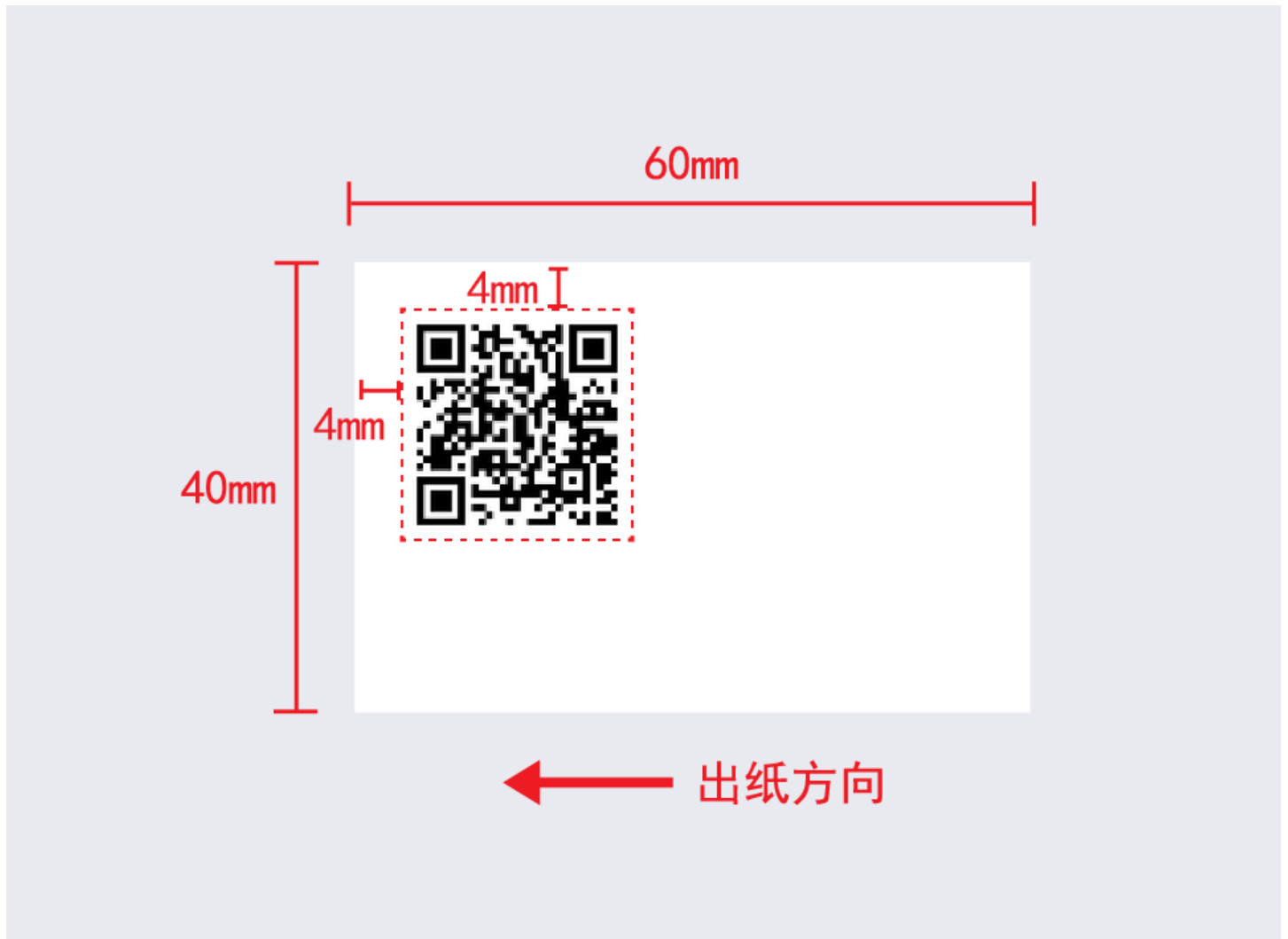
2.4 二维码绘制

代码块

```
1  /// 绘制画板
2  /// @param width 宽
3  /// @param height 高
4  /// @param horizontalShift 水平偏移,暂不可用
5  /// @param verticalShift 竖直偏移,暂不可用
6  /// @param rotate 旋转角度
7  /// @param font 使用字体路径,缺省输入nil
8  [JCAPI initDrawingBoard:60 withHeight:40 withHorizontalShift:0
  withVerticalShift:0 rotate:90 font:@""];
9  /// 绘制二维码
10 /// @param x 水平坐标
11 /// @param y 垂直坐标
12 /// @param w 标贴宽度,单位mm
13 /// @param h 标贴高度,单位mm
14 /// @param text 文本内容
15 /// @param rotate 旋转角度, 仅支持0,90,180,270
16 /// @param codeType 二维码类型 31:QR_CODE 32:PDF417 33:DATA_MATRIX 34:AZTEC
```

```
17 [JCAPI drawLableQrCode:4 withY:4 withWidth:15 withHeight:15 withString:@"二维码  
内容" withRotate:0 withCodeType:31];
```

效果图如下（文本框真实绘制效果不包含边框线）：



2.5 图形绘制

代码块

```
1  /// 绘制画板
2  /// @param width 宽
3  /// @param height 高
4  /// @param horizontalShift 水平偏移,暂不可用
5  /// @param verticalShift 竖直偏移,暂不可用
6  /// @param rotate 旋转角度
7  /// @param font 使用字体路径,缺省输入nil
8  [JCAPI initDrawingBoard:60 withHeight:40 withHorizontalShift:0
  withVerticalShift:0 rotate:90 font:@""];
```

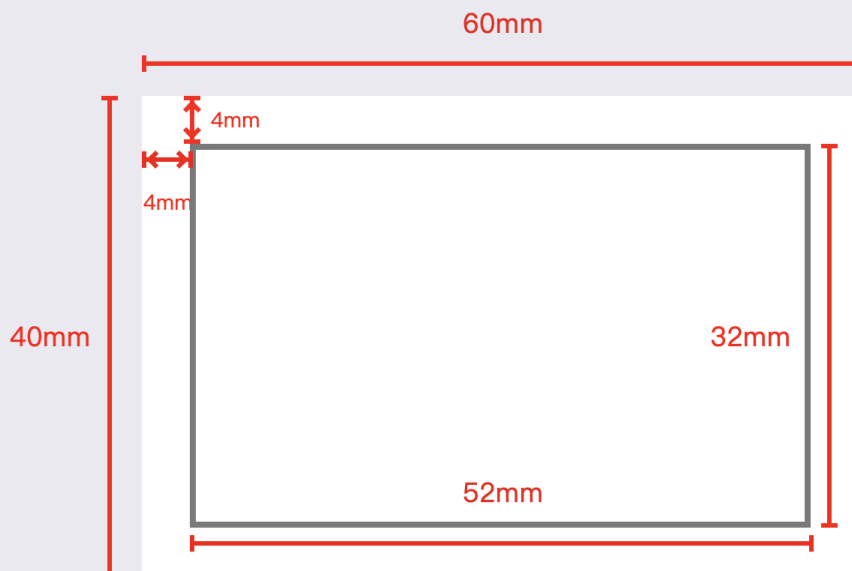


```

9    /// 绘制形状
10   /// @param x 水平坐标
11   /// @param y 垂直坐标
12   /// @param w 标贴宽度,单位mm
13   /// @param h 标贴高度,单位mm
14   /// @param lineWidth 线条类型
15   /// @param cornerRadius 图像圆角
16   /// @param rotate 旋转角度, 仅支持0,90,180,270
17   /// @param graphType 图形类型 图形类型, 1.圆 2.椭圆 3.矩形 4.圆角矩形
18   /// @param lineType 线条类型 1:实线 2:虚线类型,默认虚实比例1:1
19   /// @param dashWidth 线条为虚线宽度, 【实线段长度, 空线段长度】
20   [JCAPI DrawLableGraph:4 withY:4 withWidth:52 withHeight:32 withLineWidth:0.5
withCornerRadius:0 withRotate:0 withGraphType:3 withLineType:1 withDashWidth:
[dic valueForKey:@[1,2]]];

```

效果图如下:



2.6 图片绘制

图形绘制需要转换成base64，与其他元素类似，具体的可参考文档结合DEMO示例。如有疑问可直接咨询系统接入技术支持人员

三、标签物理尺寸与图像尺寸转换

参考: [打印机参数查询-案例编号0002](#)

四、打印图片注意事项（非LOGO）

部分用户自行生成图片进行标签打印，请注意以下事项

4.1 图片尺寸

图片尺寸需要按照标签物理尺寸转换成像素后的实际尺寸传入。非1比1比例，打印可能存在缩放，不清晰，打印不全等问题。

4.2 图片背景

部分端的SDK对于透明图片可能存在未处理的情况，导致打印黑块。部分深色背景也可能导致打印成黑块。**要求打印图片时背景设置为白底**