

# PC 端 Web 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

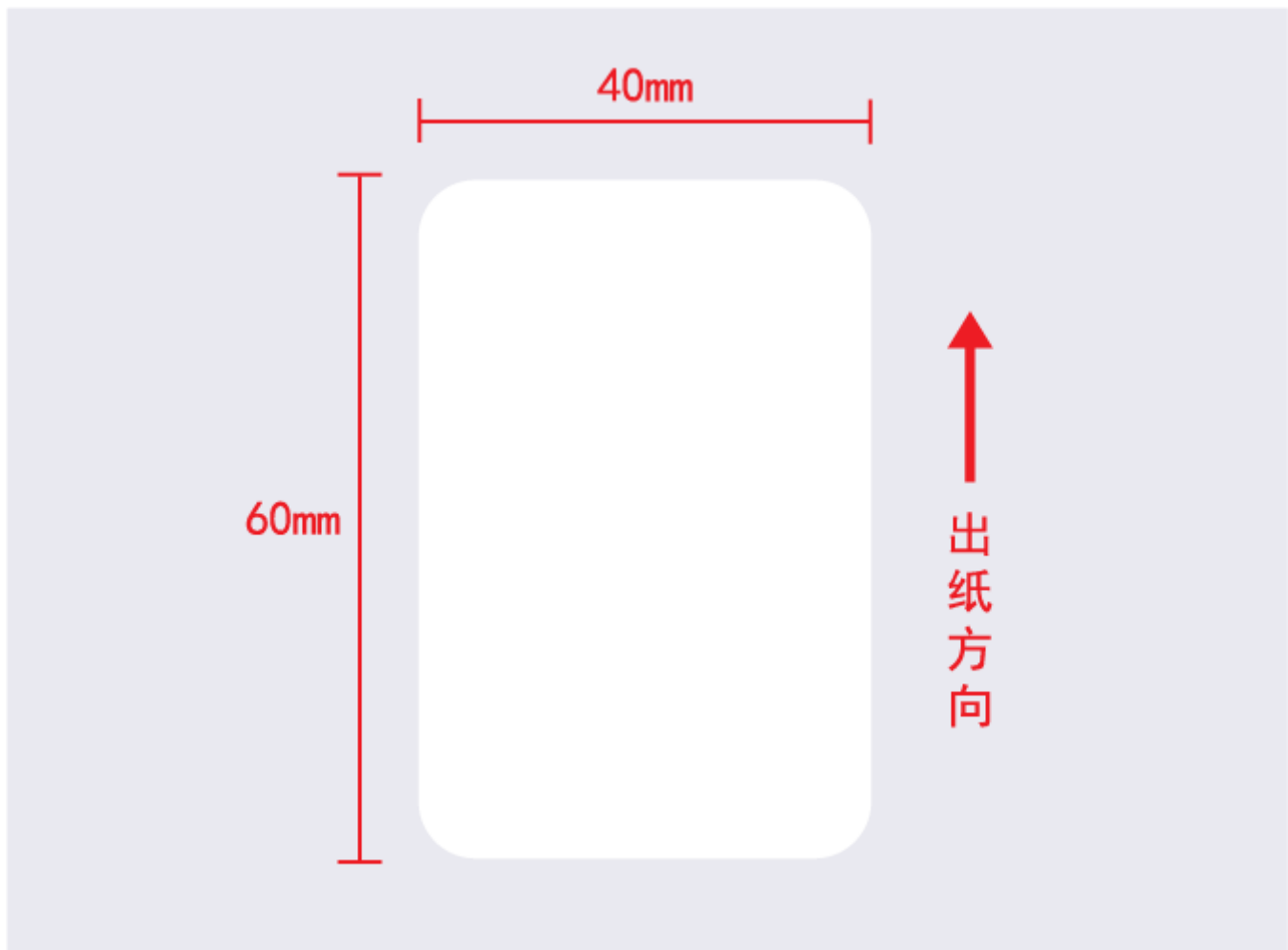
以尺寸为40-60的标签为例（单位mm）

示例代码如下：

代码块

```
1  var data = {
2    InitDrawingBoardParam: {
3      "width": 40, //画布宽度
4      "height": 60, //画布高度
5      "rotate": 0, //画布旋转角度
6      "path": "ZT001.ttf", //字体
7      "verticalShift": 0, //水平偏移, 暂不可用
8      "HorizontalShift": 0 //竖直偏移, 暂不可用
9    }
10  }
11
12  //初始化画布
13  InitDrawingBoard(data.InitDrawingBoardParam, function () {
14
15  });
```

编辑视图示例图:



## 1.2 旋转90°为例

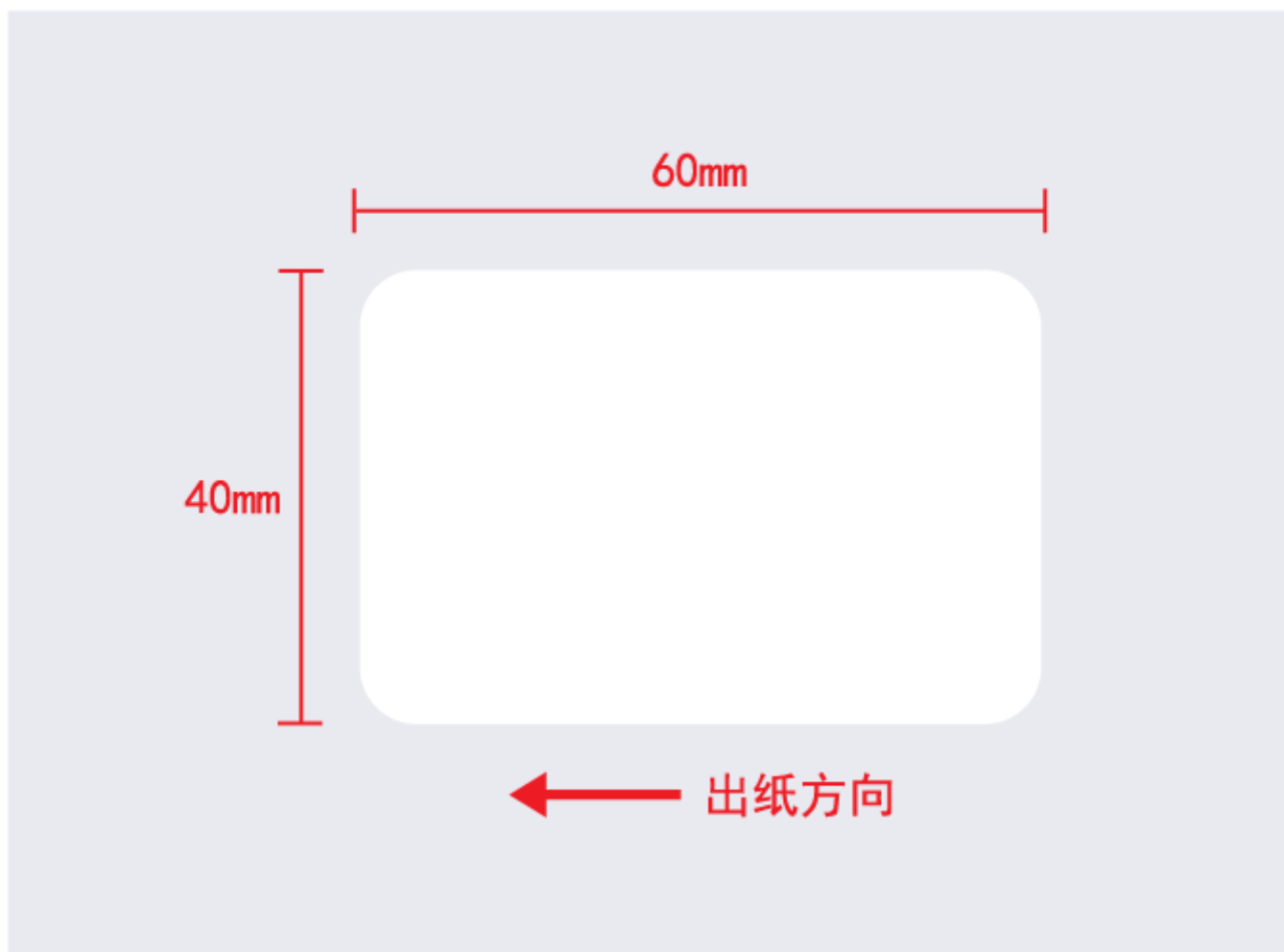
以尺寸为40-60的标签为例（单位mm）

示例代码如下：

代码块

```
1  var data = {
2    InitDrawingBoardParam: {
3      "width": 60, //画布宽度
4      "height": 40, //画布高度
5      "rotate": 90, //画布旋转角度
6      "path": "ZT001.ttf", //字体
7      "verticalShift": 0, //水平偏移, 暂不可用
8      "HorizontalShift": 0 //竖直偏移, 暂不可用
9    }
10 }
11
12 //初始化画布
13 InitDrawingBoard(data.InitDrawingBoardParam, function () {
14
15 });
```

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

示例代码如下:

代码块

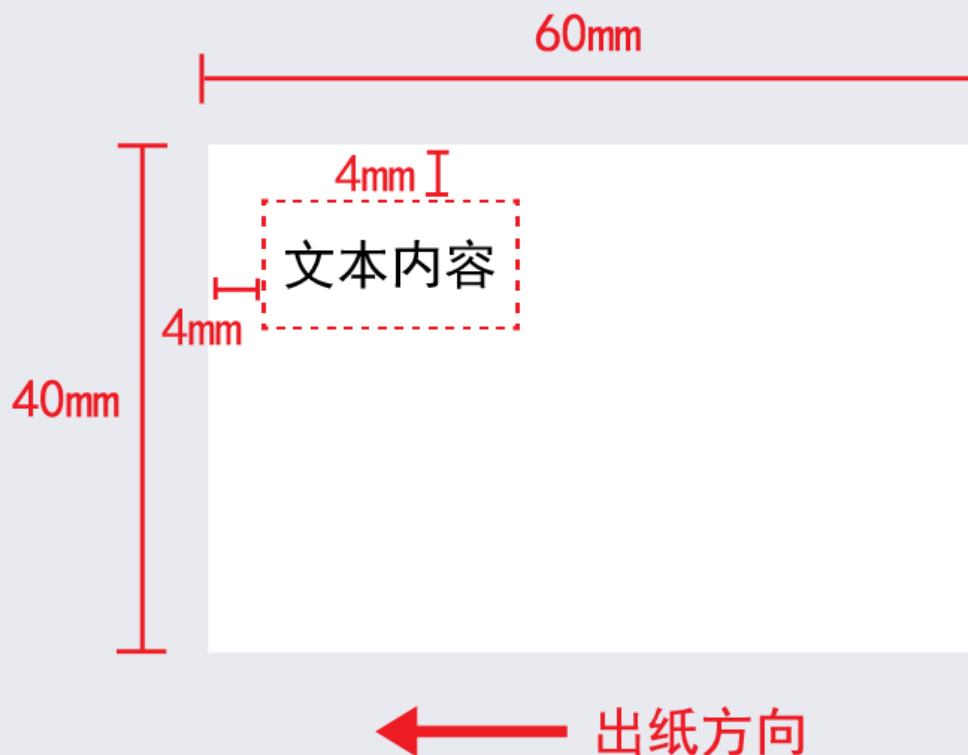
```
1  var data = {
2    InitDrawingBoardParam: {
3      "width": 60, //画布宽度
4      "height": 40, //画布高度
5      "rotate": 90, //画布旋转角度
6      "path": "ZT001.ttf", //字体
7      "verticalShift": 0, //水平偏移, 暂不可用
8      "HorizontalShift": 0 //竖直偏移, 暂不可用
```

```

9      },
10     DrawTextParam: {
11         "x": 4.0, //水平坐标
12         "y": 4.0, //垂直坐标
13         "height": 10.0, //高度, 单位mm
14         "width": 20.0, //宽度, 单位mm
15         "value": "文本内容", //绘制的文本 (UTF8)
16         "fontFamily": "", //字体, 默认使用 "ZT001.ttf"
17         "rotate": 0, //旋转角度
18         "fontSize": 3.0, //字体大小
19         "textAlignHorizonral": 1, //水平对齐方式, 0:左对齐
           1:居中 对齐 2:右对齐
20         "textAlignVertical": 1, //垂直对齐方式, 0:顶对齐 1:
           垂直居中 2:底对齐
21         "letterSpacing": 0.0, //字间距, 单位mm
22         "lineSpacing": 1.0, //行间距, 单位mm
23         "lineMode": 6, //行模式, 默认6 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)
24         "fontStyle": [false, false, false, false] //字体
           风格 (加粗、斜体、下划线), 数据格式[斜体, 加粗, 下划线, 保留]
25     }
26 }
27
28 //初始化画布
29 InitDrawingBoard(data.InitDrawingBoardParam, function () {
30     //绘制文本
31     DrawLableText(data.DrawTextParam, function () {
32
33     });
34 });

```

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

示例代码如下:

代码块

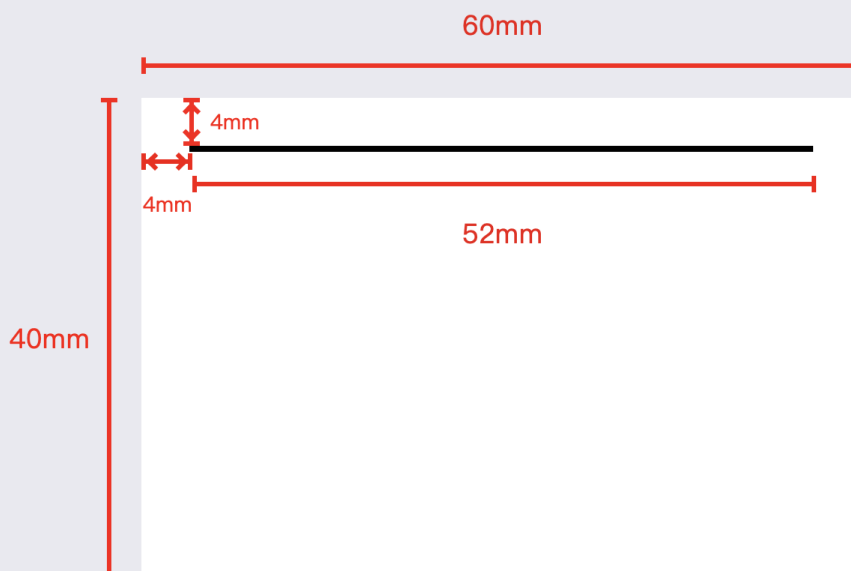
```
1  var data = {
2    InitDrawingBoardParam: {
3      "width": 60, //画布宽度
4      "height": 40, //画布高度
5      "rotate": 90, //画布旋转角度
6      "path": "ZT001.ttf", //字体
7      "verticalShift": 0, //水平偏移, 暂不可用
8      "HorizontalShift": 0 //竖直偏移, 暂不可用
9    },
10   DrawLineParam: {
11     "x": 4.0, //水平坐标
12     "y": 4.0, //垂直坐标
13     "height": 0.5, //高度, 单位mm
14     "width": 52.0, //宽度, 单位mm
15     "rotate": 0, //旋转角度
```

```

16         "lineType": 1, //线条类型, 1:实线 2:虚线
17         "dashwidth": [1,1] //虚线段长度
18     }
19 }
20
21 //初始化画布
22 InitDrawingBoard(data.InitDrawingBoardParam, function () {
23     //绘制线条
24     DrawLableLine(data.DrawLineParam, function () {
25
26     });
27
28 });

```

效果图如下:



## 2.3 一维码（条形码）绘制

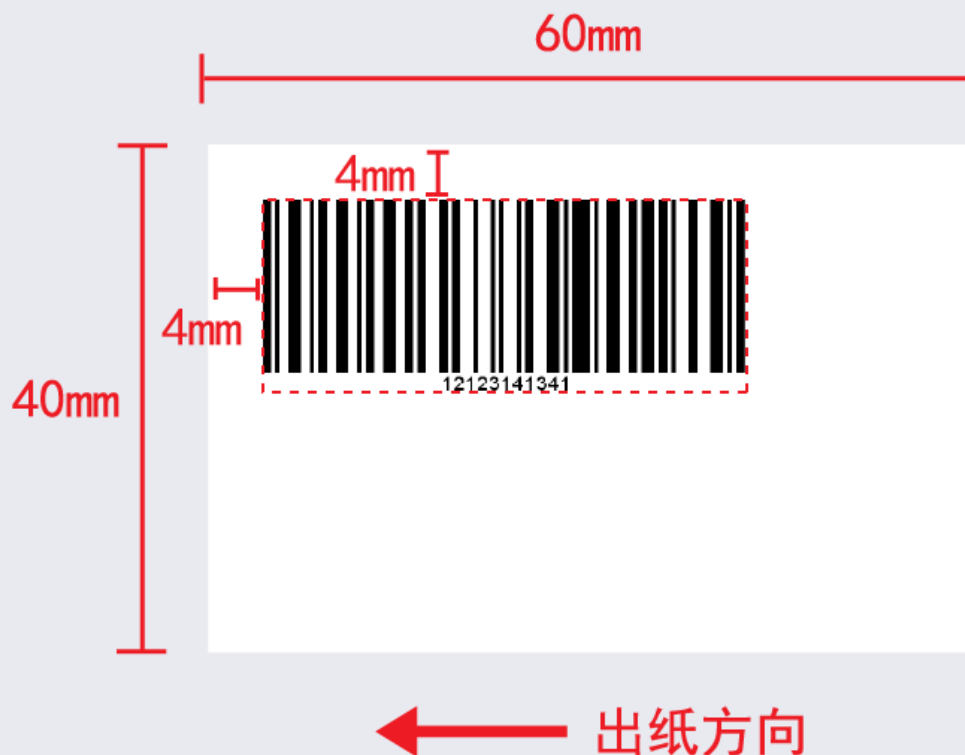
代码块

```

1  var data = {
2      InitDrawingBoardParam: {
3          "width": 60, //画布宽度
4          "height": 40, //画布高度
5          "rotate": 90, //画布旋转角度
6          "path": "ZT001.ttf", //字体
7          "verticalShift": 0, //水平偏移, 暂不可用
8          "HorizontalShift": 0 //竖直偏移, 暂不可用
9      },
10     DrawBarcodeParam: {
11         "x": 4.0, //水平坐标
12         "y": 4.0, //垂直坐标
13         "height": 15, //高度, 单位mm
14         "width": 38, //宽度, 单位mm
15         "value": '12123141341', //文本内容
16         "codeType": 20, //一维码类型20:CODE128 ,21:UPC-A,22:UPC-
17         * 25:CODE93,26:CODE39,27:CODEBAR, 28:ITF25
18         "rotate": 0, //旋转角度, 仅支持0,90,180,270
19         "fontSize": 2.0, //文本字号
20         "textHeight": 2.5, //文本高度
21         "textPosition": 0 //文本位置, int, 一维码文字识别码显示位置, 0:下方显
22         示, 1:上方显示, 2:不显示
23     }
24 }
25
26 //初始化画布
27 InitDrawingBoard(data.InitDrawingBoardParam, function () {
28     // 绘制一维码 (条形码)
29     DrawLabelBarcode(data.DrawBarcodeParam, function () {
30
31     });
32 });

```

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  var data = {
2    InitDrawingBoardParam: {
3      "width": 60, //画布宽度
4      "height": 40, //画布高度
5      "rotate": 90, //画布旋转角度
6      "path": "ZT001.ttf", //字体
7      "verticalShift": 0, //水平偏移, 暂不可用
8      "HorizontalShift": 0 //竖直偏移, 暂不可用
9    },
10   DrawQrcodeParam: {
11     "x": 4.0, //水平坐标
12     "y": 4.0, //垂直坐标
13     "height": 15, //高度, 单位mm
14     "width": 15, //宽度, 单位mm
15     "value": "二维码内容", // 文本内容
16     "rotate": 0, //旋转角度, 仅支持0, 90, 180, 270
17     "codeType": 31 //一维码类
```

型, 31: QR\_CODE, 32: PDF417, 33: DATA\_MATRIX, 34: AZTEC

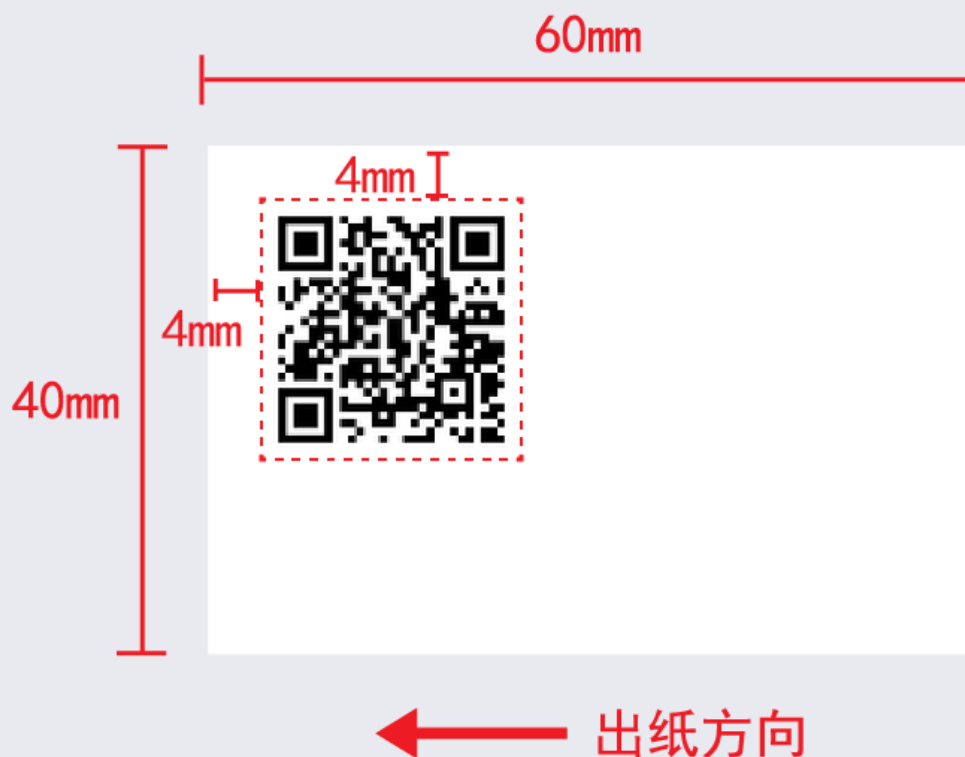


```

18         }
19     }
20
21     //初始化画布
22     InitDrawingBoard(data.InitDrawingBoardParam, function () {
23         //绘制二维码
24         DrawLabelQrCode(data.DrawQrcodeParam, function () {
25
26             });
27     });

```

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     var data = {

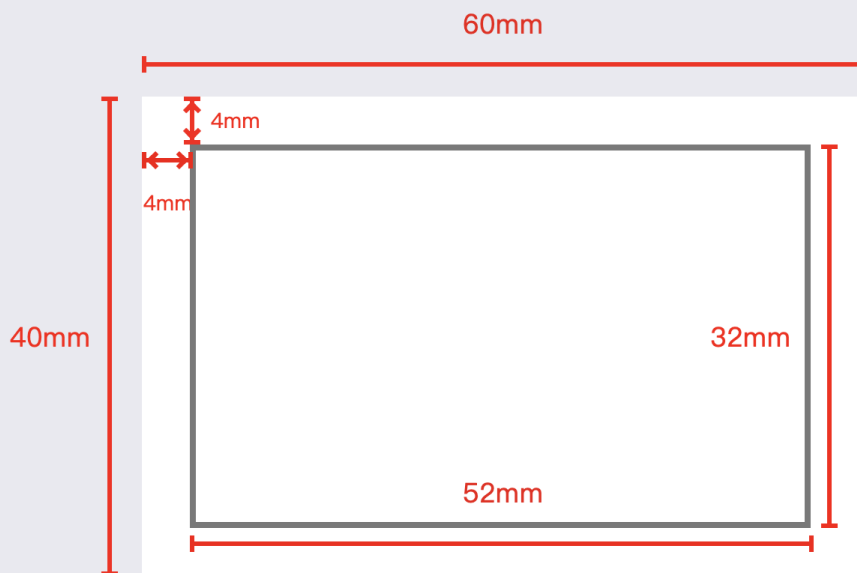
```

```

2   InitDrawingBoardParam: {
3       "width": 60, //画布宽度
4       "height": 40, //画布高度
5       "rotate": 90, //画布旋转角度
6       "path": "ZT001.ttf", //字体
7       "verticalShift": 0, //水平偏移, 暂不可用
8       "HorizontalShift": 0 //竖直偏移, 暂不可用
9   },
10  DrawGraphParam: {
11      "x": 4.0, //水平坐标
12      "y": 4.0, //垂直坐标
13      "height": 52, //宽度, 单位mm
14      "width": 32, //高度, 单位mm
15      "rotate": 0, //旋转角度, 仅支持0, 90, 180, 270
16      "graphType": 3, //图形类型, 1.矩形 2.圆角矩形 3.圆 4.椭圆
17      "cornerRadius": 0, //圆角
18      "lineWidth": 0.5, //线宽
19      "lineType": 1, //线条类型, 1:实线, 2:虚线类型, 默认虚实比例1:1
20      "dashwidth": [1, 1] //线条为虚线宽度, 【实线段长度, 空线段长度】
21  }
22  }
23
24  //初始化画布
25  InitDrawingBoard(data.InitDrawingBoardParam, function () {
26      //绘制形状
27      DrawGraphParam(data.DrawQrcodeParam, function () {
28
29          });
30  });

```

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，请注意一下事项

### 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**