

Utilizando Bitmaps para impressão v1.0.0

Introdução

1 - Exemplo de código para gerar um bitmap

- 1.1 - Criando as linhas de texto para o bitmap
- 1.3 - Criando as configurações de texto para o bitmap
- 1.4 - Calcula as medidas de texto para o bitmap
- 1.5 - Criando a imagem para o bitmap
- 1.6 - Escrevendo o texto para o bitmap

2 - Criando bitmap com View

Introdução

Este documento tem como finalidade auxiliar na criação de um bitmap para utilização do serviço de impressão. Aqui será explicado como criar um bitmap, incluir imagem, incluir texto e configurações de medidas tanto da imagem quanto do texto que serão inseridos.

1 - Exemplo de código para gerar um bitmap

No exemplo abaixo, é criada toda a estrutura do bitmap para ser gerado o comprovante dentro dos parâmetros pré-definidos.

```
private void fontDiferentBitmapClicked(){
    Vector<String> lines = createLines();
    Rect bounds = new Rect();
    //select the font and color that will be used in printing
    Paint textPaint = createTextPaint();
    //returns the measurements used to calculate text sizes
```

```

        Map<String, Integer> mapTextmeasures = calculateWidth
(lines, textPaint, bounds);
        //calculates the size of the spacing between texts
        int textHeight = mapTextmeasures.get("height");
        int textWidth = mapTextmeasures.get("width");
        //Set the logo that will be printed on the bitmap
        Bitmap image = createImage(textWidth);
        //20 adds a margin to the right of the bitmap so it d
oesn't end in the text
        int bitmapWidth = image.getWidth() + 20;
        //40 adds a space to the end of the bitmap to avoid c
utting off the text
        int bitmapHeight = image.getHeight() + textHeight + 4
0;

        //assembles the bitmap with your print settings
        Bitmap bitmap = Bitmap.createBitmap(bitmapWidth, bitm
apHeight, Bitmap.Config.ARGB_8888);
        float imageY = 10; //10 is the top margin
        createCanvas(bitmap, image, lines, bounds, imageY, te
xtPaint);
        mPrinterManager.printBitmap(bitmap);
    }

```

1.1 - Criando as linhas de texto para o bitmap

No método abaixo, é criado um vetor de Strings para incluir linha a linha os textos que serão impressos no bitmap para gerar o comprovante. Quando é inserido **lines.add("")** representa que estamos pulando uma linha para melhorar a visualização do comprovante.

```

private static Vector<String> createLines(){
    Vector<String> lines = new Vector<>();
    String cnpj = "33.666.184/0001-82";
    lines.add("");
    //used to get the current date from the machine and use it
to fill in the receipt field

```

```

LocalDateTime dateTime = LocalDateTime.now();
DateTimeFormatter formatter = DateTimeFormatter.ofPattern
("dd-MM-yyyy HH:mm:ss");

lines.add("Data:" + dateTime.format(formatter).toString());
lines.add("");
lines.add("RELATÓRIO DE VENDAS - FECHAMENTO");
lines.add("");
lines.add("POSITIVO 2024");
lines.add("");
lines.add("CNPJ:" + cnpj);
lines.add("");
lines.add("");
lines.add("TERMINAL:" + 159753654);
lines.add("");
lines.add("-");
lines.add("");
lines.add("DE: 01/01/2024 00:00:00");
lines.add("");
lines.add("ATE: 16/01/2024 00:00:00");
lines.add("");
lines.add("STATUS                VENDAS                VALO
R");
lines.add("");
lines.add("APROVADAS                0002                3,0
0");
lines.add("");
lines.add("CANCELADAS(-)            0000                0,0
0");
lines.add("");
lines.add("TOTAL                0002                3,0
0");
lines.add("");
lines.add("DEBITO                0001                1,5
0");
lines.add("");

```

```

        lines.add("CREDITO                                0001                1,5
0");
        lines.add("");
        lines.add("TOTAL                                0002                3,0
0");
        lines.add("");
        lines.add("PIX-----");
        lines.add("");
        lines.add("PIX DEB A VISTA                        0002                3,0
0");
        lines.add("");
        lines.add("TOTAL                                0002                3,0
0");
        lines.add("");
        lines.add("***** MENSAGEM DE COMPROVANTE *****");
        lines.add("");
        lines.add("SAC:0800000000 OUVIDORIA:0800000000");
        return lines;
    }

```

1.3 - Criando as configurações de texto para o bitmap

A seguir informamos qual será a fonte utilizada e a cor da fonte:

Typeface.createFromAsset(getAssets(), "stocky.ttf")

```

//create print settings that will generate font, text color
private Paint createTextPaint(){
    Typeface typeface = Typeface.createFromAsset(getAssets(),
"DejaVuSansMono-Bold_0.ttf");
    Paint textPaint = new Paint();
    textPaint.setColor(Color.BLACK);
    textPaint.setTypeface(typeface);
    return textPaint;
}

```

1.4 - Calcula as medidas de texto para o bitmap

Nesse método, é feito o cálculo dos tamanhos de largura e altura do texto para garantir que todo o texto seja escrito dentro do bitmap sem ocorrer cortes na hora da impressão.

```
//method used to calculate the size of texts that will be inserted into the bitmap
private Map<String, Integer> calculateWidth(Vector<String> lines, Paint textPaint, Rect bounds){
    int textHeight = 0;
    int textWidth = 0;
    Map<String, Integer> results = new HashMap<>();
    for(String line: lines){
        textPaint.getTextBounds(line, 0, line.length(), bounds);
        textHeight += bounds.height();
        float width = textPaint.measureText(line);
        if(width > textWidth){
            textWidth = (int) width;
        }
    }
    //performs the calculation to ensure that the height of the bitmap does not cut off the text
    //-20 value can be adjusted according to need
    int lineHeight = bounds.height();
    textHeight += (lines.size()- 20) * lineHeight;
    results.put("height", textHeight);
    results.put("width", textWidth);
    return results;
}
```

1.5 - Criando a imagem para o bitmap

Aqui iremos informar para o

BitmapFactory.decodeResource(mContext.getResources(), R.drawable.positivo)
qual vai ser a imagem utilizada para inserção dentro do bitmap e passaremos também

as configurações de medida da imagem para garantir que ela não fique grande demais usando a medida de `textWidth` como parâmetro de tamanho.

```
//method used to search for the image within the drawable folder and calculate the size of the image according to the width that will be used for the text
//to ensure that the image does not exceed the size of the printed text
private Bitmap createImage(float textWidth){
    Bitmap bitmapImageOriginal = BitmapFactory.decodeResource(mContext.getResources(), R.drawable.positivo);
    int newImageWidth = (int) textWidth;
    int newImageHeight = (int) ((float) bitmapImageOriginal.getHeight() * ((float) newImageWidth / (float) bitmapImageOriginal.getWidth()));
    Bitmap resizedImage = Bitmap.createScaledBitmap(bitmapImageOriginal, newImageWidth, newImageHeight, true);
    return resizedImage;
}
```

1.6 - Escrevendo o texto para o bitmap

Neste ponto, utilizamos o Canvas para escrever as informações passadas para dentro do bitmap, inserindo a imagem tratada e o texto escrito dentro de **lines** linha a linha para garantir que todo o texto seja escrito.

```
//creates the canvas that will draw the text inside the bitmap
private void createCanvas(Bitmap bitmap, Bitmap image, Vector<String> lines, Rect bounds, float imageY, Paint textPaint){
    Canvas canvas = new Canvas(bitmap);
    canvas.drawColor(Color.WHITE);
    float centralize = (bitmap.getWidth() - image.getWidth()) / 2f; //calculation used to center the image in the bitmap
}
```

```

        canvas.drawBitmap(image, centralize, imageY, textPaint);
    }
    //Include the text below the image so as not to overlap one another
    float baseline = image.getHeight() + 40;
    for(String line : lines){
        canvas.drawText(line, 10, baseline , textPaint);
        baseline += bounds.height();
    }
}

```

2 - Criando bitmap com View

Neste exemplo é utilizado uma view para gerar o modelo de comprovante que sera passado para o bitmap, aqui é incluso o alinhamento do texto, fonte e tamanho do texto utilizado para impressão.

```

private void onViewToBitmapClicked() {
    final Bitmap bitmapText = generateBitmapFromString();
    final Map<String,Integer> attr = new HashMap<String,Integer>();
    attr.put(PrinterManager.KEY_ALIGN, 0);
    attr.put(PrinterManager.KEY_TYPEFACE, 1);
    attr.put(PrinterManager.KEY_TEXTSIZE, 17);
    ThreadPoolManager.getInstance().executeTask(() -> {
        try {
            mPrinterManager.printBitmap(bitmapText);
            mPrinterManager.printBase64Bitmap(bitmapText, attr);
        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}

```

Este método é responsável por buscar o arquivo comprovante.xml que contém o modelo de comprovante, passar suas configurações de medidas e customizações através da criação de uma view e escrever o texto dentro do bitmap.

```
private Bitmap generateBitmapFromString() {
    View view = LayoutInflater.from(this).inflate(R.layout
t.comprovante, null);

    int targetWidth = 500;
    int targetHeight = 1100;
    // Medir e layoutizar a view
    int widthMeasureSpec = View.MeasureSpec.makeMeasureSpec
ec(targetWidth, View.MeasureSpec.EXACTLY);
    int heightMeasureSpec = View.MeasureSpec.makeMeasures
pec(targetHeight, View.MeasureSpec.EXACTLY);
    view.measure(widthMeasureSpec, heightMeasureSpec);
    view.layout(0, 0, view.getMeasuredWidth(), view.getMe
asuredHeight());

    Bitmap bitmap = Bitmap.createBitmap(targetWidth, targ
etHeight, Bitmap.Config.RGB_565);
    Canvas canvas = new Canvas(bitmap);

    // Limpar o canvas com uma cor de fundo
    canvas.drawColor(Color.WHITE);

    view.draw(canvas);
    return bitmap;
}
```

Aqui é um exemplo de código do XML que foi usado para gerar o comprovante passado para o bitmap.

```
<TextView
    android:id="@+id/titleTextView"
    android:layout_width="match_parent"
```



```
android:layout_height="wrap_content"
android:text="CREDITO A VISTA - I"
android:textSize="20sp"
android:textStyle="bold" />
```

<View

```
android:layout_width="match_parent"
android:layout_height="1dp"
android:background="#000000" />
```

<LinearLayout

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical">
```

<TextView

```
android:textSize="10sp"
android:id="@+id/cnpjTextView"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="CNPJ: 00.000.000/000000" />
```

<TextView

```
android:textSize="10sp"
android:id="@+id/nomeTextView"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="NOME: TESTE LTDA" />
```

<TextView

```
android:textSize="10sp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="END: RUA TESTE, 000 TESTE" />
```

<TextView

```

        android:textSize="10sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="IE: TESTE" />
<TextView
    android:textSize="10sp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="IM: 0000000000" />
<TextView
    android:textSize="10sp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="DATA: 01/01/2016 HORA: 00:00:00" />
<TextView
    android:textSize="10sp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="CUPOM: 000000 C00: 000000" />
<TextView
    android:textSize="10sp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="ITEM CODIGO DESCRICAO QTD UN VL UNIT(R$)"
<TextView
    android:textSize="10sp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="VISA CREDITO 1 1,00 1,00 ***** 1,00" />
</LinearLayout>

```

Abaixo a imagem que representa essa View usada no XML acima:

CREDITO A VISTA - I

CNPJ: 00.000.000/000000
NOME: TESTE LTDA
END: RUA TESTE, 000 TESTE
IE: TESTE
IM: 000000000
DATA: 01/01/2016 HORA: 00:00:00
CUPOM: 000000 COD: 000000
ITEM CODIGO DESCRICAO QTD UN VL UNIT(R\$)
VISA CREDITO 1 1,00 1,00 ***** 1,00
[*****]*****008989988888-0
1000000090938891009947821009381969379123
store 00000000

VISA CREDITO 1 1,00 1,00 ***** 1,00
1 | 00000000
store 00000000

VISA CREDITO 1 1,00 1,00 ***** 1,00
1 | 00000000
store 00000000
VISA CREDITO 1 1,00 1,00
VISA - ***** 0090

VISA CREDITO 1 1,00 1,00 ***** 1,00
1 | 00000000
store 00000000