About This Article

This article explains about creating well formatted tables using Table Layout in Android. Sample code snippets are also provided for practice and better understanding.

Scope:

This article explains on using Table Layout while developing Applications in Android. It assumes that the user has already installed Android and necessary tools to develop application. It also assumes that the user to be familiar with Java

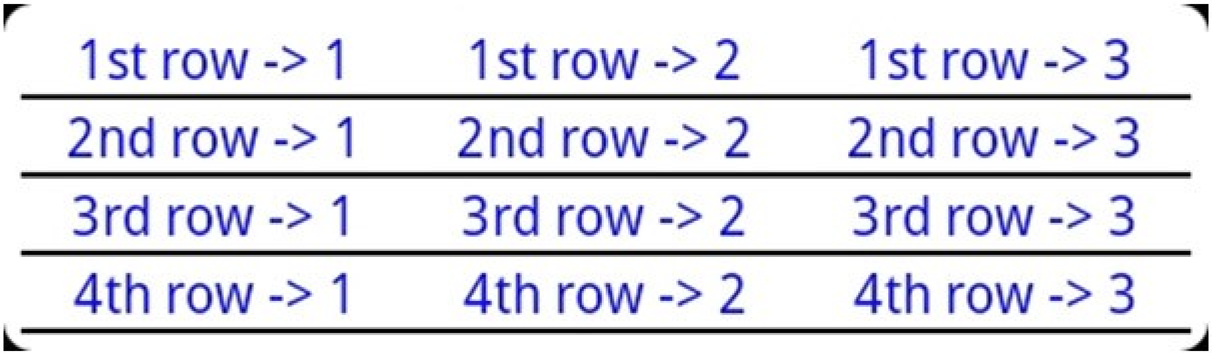
Introduction

In this article, we are going to present the methods available to create well formatted tables using Table Layout. In general, you should use XML files to create layouts, but sometimes it's not possible to create a layout with XML files without managing it directly from the source code.
We'll present a few ways to build tables using XML, showing you the most important things you might want to do with your table. Also, we'll show you what to do when you cannot prepare your table in XML, as you'll have to do this dynamically from code.

Creating Table from XML

A well known issue is that there are no parameters that automatically setup borders or lines between rows in TableLayout. However there i's a simple way to do this. Let's assume we want a white table (e.g. white background and blue font) with black lines between rows. The effect we want to achieve is presented in Figure 1. Everything can be done in the XML layout file. At first we need to describe the Table Layout:

```
<TableLayout
         android:id="@+id/simple_table"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:background="#FFF"
         android:stretchColumns="0" >
         % (1) Here we put the content of the table
</TableLayout>
```

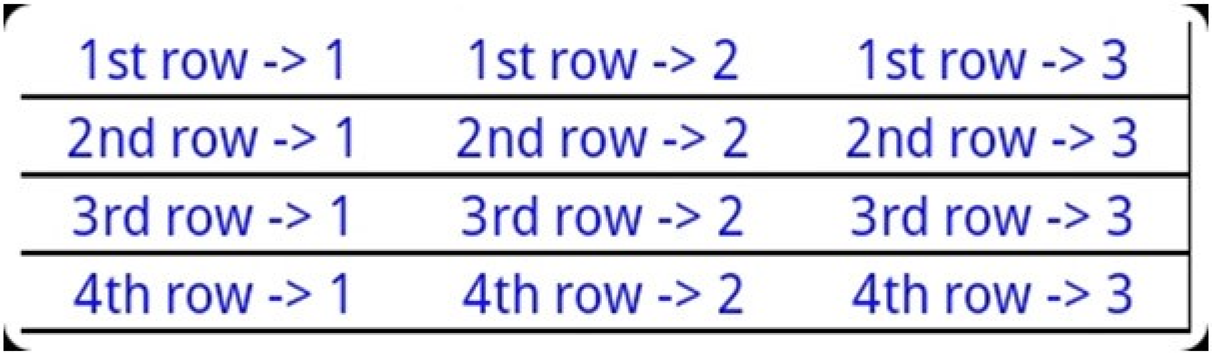


1st row -> 1	1st row -> 2	1st row -> 3
2nd row -> 1	2nd row -> 2	2nd row -> 3
3rd row -> 1	3rd row -> 2	3rd row -> 3
4th row -> 1	4th row -> 2	4th row -> 3

The most important thing is that the background color is set to white. If the layout width is set to fill_parent, then the stretch columns parameter should be set to at least one column. Otherwise, the table will be stretched to fill all the available space, but the columns may not necessarily fill the whole table and there will be free space (in the color of the background row) visible on the right (or left) side of the table. We can also prepare our own table background shape. We’ve put this into the res/drawable/simple_table_background.xml file.

Another issue with the parameter ‘layout width’ set to fill parent: If we have more than one column and they are set to stretch, it can happen that there is one pixel left, and more than one column will want to acquire it. In this situation, none of the columns will get it, and this pixel will remain in the color of the background row. This issue is presented in Figure 2. The location of this one-pixel-column depends on the gravity – maybe right or left (it might even appear on both sides at the same time, if there's more than one free pixel).

```
<TableLayout
         android:id="@+id/simple_table"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:background="#FFF"
         android:stretchColumns="0" >
         % (1) Here we put the content of the table
</TableLayout>
```

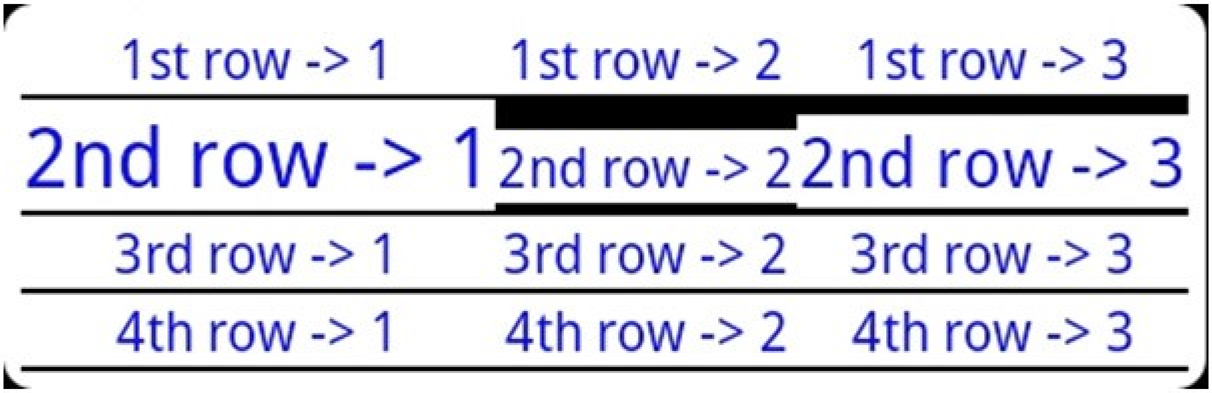


1st row -> 1	1st row -> 2	1st row -> 3
2nd row -> 1	2nd row -> 2	2nd row -> 3
3rd row -> 1	3rd row -> 2	3rd row -> 3
4th row -> 1	4th row -> 2	4th row -> 3

After we create the background shape, we can change android:background="#FFF" to android:background= "@drawable/simple_table_background" in the TableLayout file. This will cause our table layout to have a white background and rounded corners. In (1) we input the table rows (as many as we want) and every row should have a black background (it doesn’t have to be black, it can be in the color we want the lines between the rows to have).

Inside every row we put a Text View – two parameters are important: the background color, which should be the same as the table layout background color (white in this case), and the marginBottom. If the bottom margin is set, then we can see part of a table row (which is black) below the Text View, so it'll look like a line below this row.

You should notice that the height of all cells in a single row is measured and the next row appears after the cell with the lowest position. So, if cells have different sizes in the same row, all but the biggest one will be positioned with a margin in the color of the background. This should be taken into account when using different fonts. To avoid this issue, you may set the layout_height as a constant value (e.g. 30dp) for all cells in the row. This issue is presented in Figure 3.

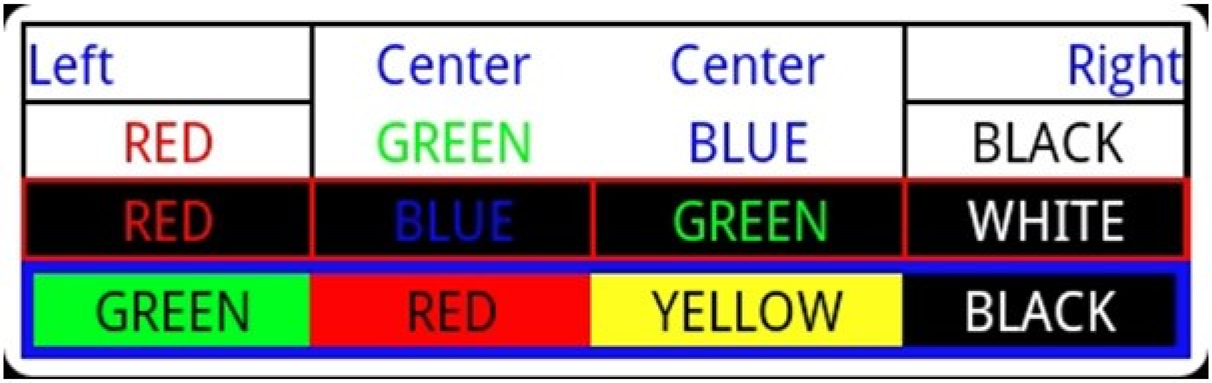


1st row -> 1	1st row -> 2	1st row -> 3
2nd row -> 1	2nd row -> 2	2nd row -> 3
3rd row -> 1	3rd row -> 2	3rd row -> 3
4th row -> 1	4th row -> 2	4th row -> 3

```
<TableRow
         android:id="@+id/tableRow1"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:background="#000" >
         <TextView
                  android:id="@+id/textView1"
                  android:background="#FFF"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
                  android:text="Some text...”
                  android:textColor="#00F"
                  android:layout_marginBottom="1dp">
         </TextView>
         % More Text View in this row
</TableRow>
```

We can create a line above every row and lines on the left and right side of every column (by setting the corresponding margins) in the same way. Notice that when you set the right margin for the first column to 1dp and the left margin for the second column to 1dp you will get a vertical line 2dp thick between them (the same with the top/bottom margins).

You have to set the background color for every row separately, so lines in all rows can have different colors. There's no way to draw lines between columns in the same row in different colors using this method.



Left	Center	Center	Right
RED	GREEN	BLUE	BLACK
RED	BLUE	GREEN	WHITE
GREEN	RED	YELLOW	BLACK

Sample Example

The following example shows you how to create the Table Layout from the XML file. This application is called FullTable. Also, this app presents the layout described above – you can choose which layout you want to see. It probably presents all capabilities of the described method. You can see the effect of this source in Figure 4.

FullTableActivity.java

```
package test.layout;

import android.app.Activity;
import android.os.Bundle;

public class FullTableActivity extends Activity {

        /** Called when the activity is first created. */
        @Override
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                // You can choose one of the layouts you want to see
// You can also modify the layout by making changes
// in the main.xml or main_2.xml files.

                setContentView(R.layout.main);
                // setContentView(R.layout.main_2);

        }

}
```

Add below main.xml or main_2.xml in your project res/layout/

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/simple_table"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/simple_table_background"
        android:stretchColumns="0,1,2,3" >
<TableRow
        android:id="@+id/tableRow1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#000" >
        <TextView
                android:id="@+id/textView1a"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="Left"
                android:textColor="#00F"
                android:layout_marginBottom="1dp"
                android:layout_marginTop="1dp"
                android:layout_marginLeft="1dp">
        </TextView>
        <TextView
                android:id="@+id/textView2a"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="Center"
                android:textColor="#00F"
                android:layout_marginTop="1dp"
                android:layout_marginLeft="1dp"
                android:paddingBottom="1dp"
                android:gravity="center" >
        </TextView>
        <TextView
                android:id="@+id/textView3a"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="Center"
                android:textColor="#00F"
                android:layout_marginTop="1dp"
                android:paddingBottom="1dp"
                android:gravity="center" >
        </TextView>
        <TextView
                android:id="@+id/textView4a"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="Right"
                android:textColor="#00F"
                android:layout_marginBottom="1dp"
                android:layout_marginTop="1dp"
                android:layout_marginLeft="1dp"
                android:layout_marginRight="1dp"
                android:gravity="right" >
        </TextView>
</TableRow>

<TableRow
        android:id="@+id/tableRow2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#000" >
        <TextView
                android:id="@+id/textView1b"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="RED"
                android:textColor="#F00"
                android:layout_marginLeft="1dp"
                android:gravity="center" >
        </TextView>
        <TextView
                android:id="@+id/textView2b"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="GREEN"
                android:textColor="#0F0"
                android:layout_marginLeft="1dp"
                android:gravity="center" >
        </TextView>
        <TextView
                android:id="@+id/textView3b"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="BLUE"
                android:textColor="#00F"
                android:gravity="center" >
        </TextView>
        <TextView
                android:id="@+id/textView4b"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="BLACK"
                android:textColor="#000"
                android:layout_marginLeft="1dp"
                android:layout_marginRight="1dp"
                android:gravity="center" >
        </TextView>
</TableRow>

<TableRow
        android:id="@+id/tableRow3"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#f00" >
        <TextView
                android:id="@+id/textView1c"
                android:background="#000"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="RED"
                android:textColor="#F00"
                android:layout_marginTop="1dp"
                android:layout_marginBottom="1dp"
                android:layout_marginLeft="1dp"
                android:gravity="center" >
        </TextView>
        <TextView
                android:id="@+id/textView2c"
                android:background="#000"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="BLUE"
                android:textColor="#00F"
                android:layout_marginTop="1dp"
                android:layout_marginBottom="1dp"
                android:layout_marginLeft="1dp"
                android:gravity="center" >
        </TextView>
        <TextView
                android:id="@+id/textView3c"
                android:background="#000"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="GREEN"
                android:textColor="#0F0"
                android:layout_marginTop="1dp"
                android:layout_marginBottom="1dp"
                android:layout_marginLeft="1dp"
                android:gravity="center" >
        </TextView>
        <TextView
                android:id="@+id/textView4c"
                android:background="#000"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="WHITE"
                android:textColor="#FFF"
                android:layout_marginTop="1dp"
                android:layout_marginBottom="1dp"
                android:layout_marginLeft="1dp"
                android:layout_marginRight="1dp"
                android:gravity="center" >
        </TextView>
</TableRow>
```

```
<TableRow
         android:id="@+id/tableRow3"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:background="#00F" >
         <TextView
                  android:id="@+id/textView1c"
                  android:background="#0F0"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
                  android:text="GREEN"
                  android:textColor="#000"
                  android:layout_marginTop="3dp"
                  android:layout_marginBottom="3dp"
                  android:layout_marginLeft="3dp"
                  android:gravity="center" >
         </TextView>
         <TextView
                  android:id="@+id/textView2c"
                  android:background="#F00"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
                  android:text="RED"
                  android:textColor="#000"
                  android:layout_marginTop="3dp"
                  android:layout_marginBottom="3dp"
                  android:gravity="center" >
         </TextView>
         <TextView
                  android:id="@+id/textView3c"
                  android:background="#FF0"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
                  android:text="YELLOW"
                  android:textColor="#000"
                  android:layout_marginTop="3dp"
                  android:layout_marginBottom="3dp"
                  android:gravity="center" >
         </TextView>
```

```
        <TextView
                android:id="@+id/textView4c"
                android:background="#000"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="BLACK"
                android:textColor="#FFF"
                android:layout_marginTop="3dp"
                android:layout_marginBottom="3dp"
                android:layout_marginRight="3dp"
                android:gravity="center" >
        </TextView>
</TableRow>
</TableLayout>
```

main_2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/simple_table"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/simple_table_background"
        android:stretchColumns="0,1,2"
xmlns:android1="http://schemas.android.com/apk/res/android">
<TableRow
        android:id="@+id/tableRow1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#000" >
        <TextView
                android:id="@+id/textView1a"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="1st row -> 1"
                android:textColor="#00F"
                android:layout_marginBottom="1dp"
                android:gravity="center" >
        </TextView>
        <TextView
                android:id="@+id/textView2a"
                android:background="#FFF"
```

```
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
                  android:text="1st row -> 2"
                  android:textColor="#00F"
                  android:layout_marginBottom="1dp"
                  android:gravity="center">
         </TextView>
         <TextView
                  android:id="@+id/textView3a"
                  android:background="#FFF"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
                  android:text="1st row -> 3"
                  android:textColor="#00F"
                  android:layout_marginBottom="1dp"
                  android:gravity="center">
         </TextView>
</TableRow>

<TableRow
         android:id="@+id/tableRow2"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:background="#000" >
         <TextView
                  android:id="@+id/textView1b"
                  android:background="#FFF"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
                  android:text="2nd row -> 1"
                  android:textColor="#00F"
                  android:layout_marginBottom="1dp"
                  android:gravity="center" >
         </TextView>
         <TextView
                  android:id="@+id/textView2b"
                  android:background="#FFF"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
```

```
                android:text="2nd row -> 2"
                android:textColor="#00F"
                android:layout_marginBottom="1dp"
                android:gravity="center" >
        </TextView>
        <TextView
                android:id="@+id/textView3b"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="2nd row -> 3"
                android:textColor="#00F"
                android:layout_marginBottom="1dp"
                android:gravity="center" >
        </TextView>
</TableRow>

<TableRow
        android:id="@+id/tableRow3"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#000" >
        <TextView
                android:id="@+id/textView1c"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="3rd row -> 1"
                android:textColor="#00F"
                android:layout_marginBottom="1dp"
                android:gravity="center" >
        </TextView>
        <TextView
                android:id="@+id/textView2c"
                android:background="#FFF"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceSmall"
                android:text="3rd row -> 2"
                android:textColor="#00F"
                android:layout_marginBottom="1dp"
```

```
                  android:gravity="center" >
         </TextView>
         <TextView
                  android:id="@+id/textView3c"
                  android:background="#FFF"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
                  android:text="3rd row -> 3"
                  android:textColor="#00F"
                  android:layout_marginBottom="1dp"
                  android:gravity="center" >
         </TextView>
</TableRow>

<TableRow
         android:id="@+id/tableRow3"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:background="#000" >
         <TextView
                  android:id="@+id/textView1c"
                  android:background="#FFF"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
                  android:text="4th row -> 1"
                  android:textColor="#00F"
                  android:layout_marginBottom="1dp"
                  android:gravity="center" >
         </TextView>
         <TextView
                  android:id="@+id/textView2c"
                  android:background="#FFF"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
                  android:text="4th row -> 2"
                  android:textColor="#00F"
                  android:layout_marginBottom="1dp"
                  android:gravity="center" >
         </TextView>
         <TextView
```

```
                  android:id="@+id/textView3c"
                  android:background="#FFF"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:textAppearance="?android:attr/textAppearanceSmall"
                  android:text="4th row -> 3"
                  android:textColor="#00F"
                  android:layout_marginBottom="1dp"
                  android:gravity="center" >
         </TextView>
</TableRow>

</TableLayout>
```

Creating Table from the source code

Sometimes we can't predict how many rows the table will have and there's no possibility to create it in the XML file. For example: We get a list of people from the server and we want to write every person in a single row. At the beginning, we have no idea how many people we will get and it is possible to get no people at all (the list may be empty), so there's no way to create a XML file with the appropriate layout. In this case, we have to create the layout dynamically in the source code. Of course, a large part of this layout should be prepared in the XML file. In our source code we will only create rows and fill them with the names we get from the server.

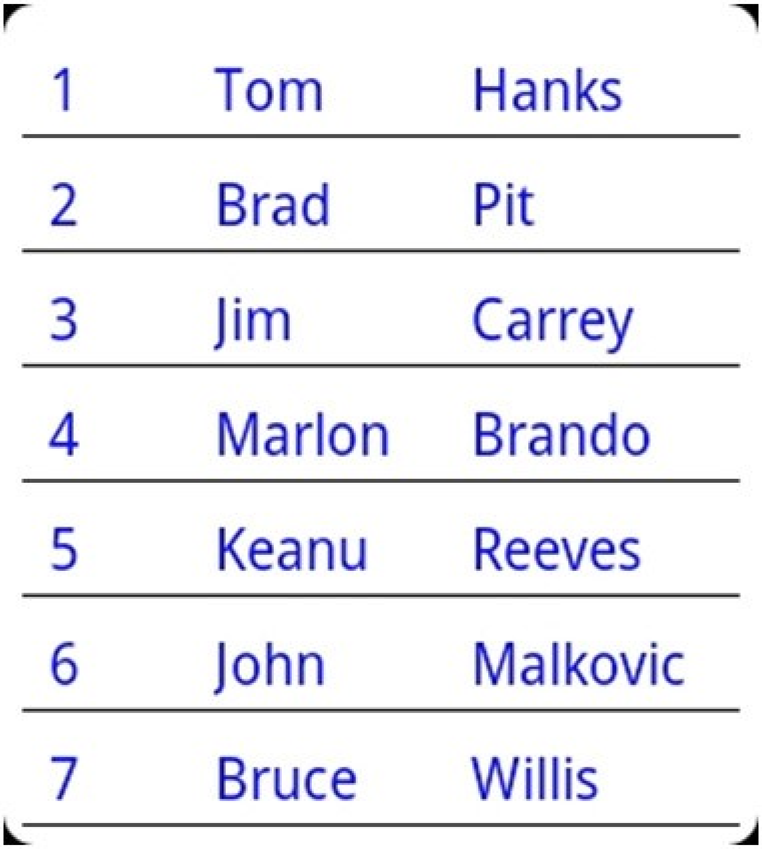
Let’s assume that we downloaded data from the server earlier, and now we have two tables of Strings: firstNames, lastNames. The sample application is quite simple – the overridden onCreate() method calls the fillTable() method – which calls the fillRow(...) method for every row we want to create and fills it with some names. The most important is the fillRow(...) method – there are three sections to setup: the number, the first name and the last name in a single row. These sections look pretty much the same, so we describe only one – for example the first name section.

```
TextView firstN = new TextView(this);
firstN.setBackgroundColor(Color.WHITE);
firstN.setTextColor(Color.BLUE);
firstN.setText(firstNames[noRow]);
row.addView(firstN);
llp = (LinearLayout.LayoutParams) firstN.getLayoutParams();
llp.setMargins(0, 0, 0, 1);
firstN.setLayoutParams(llp);
firstN.setPadding(10, 10, 20, 3);
```

The first four lines create a text viewing field and set up its basic parameters such as colors and text. In the next line, this text view is added to the currently created row (the row is a view type TableRow passed to the fillRow(...) method by arguments).

In the next three lines we are getting the layout parameters from this text view, adding the bottom margin (1 px), and applying this layout parameter to this text view. It's very important to add this text view to the row before and after to get and change the layout parameters – in other cases, the getLayoutParams() method will return null and setting up any parameters (like margins) in this view will throw an exception.

It is important to remember that when you create any view from code by calling its constructor, before you change any layout parameters in this view, you have to add this view to some parent. Padding in the last line of this code is for aligning columns – it's not necessary, but it's to make the table look better. You can see what this looks like in Figure 5.



1	Tom	Hanks
2	Brad	Pit
3	Jim	Carrey
4	Marlon	Brando
5	Keanu	Reeves
6	John	Malkovic
7	Bruce	Willis

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/scrollView1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

        <TableLayout
        android:id="@+id/tablelayout"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/table_background"
        android:shrinkColumns="0" >

          <TableRow
```

```
                  android:id="@+id/tableRow1"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                android:background="#000" >
           </TableRow>

         </TableLayout>

</ScrollView>
```

You can find the source code of this sample application (called SampleTable) at the bottom. This application also contains a second implemented method to create this table, described below.

The second way to create that table is to use a Layout Inflater. The result looks exactly the same as in the previous method – you can see it in Figure 5. From our main.xml file we have to remove the Table Row section, create a separate row.xml file and put this file into the res/layout/ folder.

In this XML file, we have to add as many text views as we need in a single row, and setup all parameters for them such as paddings, margins, colors, etc. – all parameters without text (we can add any text here, but later it'll be changed so this doesn’t make any sense). In other words – we have to prepare every single row of our table. This is good because we don't have to change and set any parameters of the view from the source code (except the text). Of course you can change any parameter while adding a row.

Also, if you plan on changing values in those text views in the future, you should set a unique ID for each view. Otherwise, all text views in a single column will have the same ID, and the method View.findViewById(...) will only return views from the first row. However, if you don't want to change the parameters of one or a few single text views, you don't need to set IDs for views. Also, you can create the whole table again and put new values to views.

The FillTable() function should look like this:

```
public void fillTable () {
         int rowCount = firstNames.length;
         Log.d("Fill Table", "rowCount = "+rowCount);
         TableLayout table = (TableLayout) this.findViewById(R.id.tablelayout);
         for(int i=0;i<rowCount;i++){
                  fillRow2(table, i);
         }
}
```

The FillRow2(...) method is a new function which fills rows in the table using the Layout Inflater. You should notice that the fillRow2(...) method takes Table View (not Row View like in fillTable(...)) and an integer which is the number of rows to fill (like before) as parameters. FillRow2(...) looks like this:

```
public void fillRow2(TableLayout table, int noRow){
         LayoutInflater inflater = (LayoutInflater)
         getSystemService(Context.LAYOUT_INFLATER_SERVICE);
         View fullRow = inflater.inflate(R.layout.row, null);
```

```
        TextView nr = (TextView) fullRow.findViewById(R.id.nr);
        nr.setText(String.valueOf(noRow+1));
        TextView fName = (TextView) fullRow.findViewById(R.id.fName);
        fName.setText(firstNames[noRow]);
        TextView lName = (TextView) fullRow.findViewById(R.id.lName);
        lName.setText(lastNames[noRow]);
        table.addView(fullRow);
}
```

This method can be used when you have to produce very complicated tables, with many parameters set differently from the default. In XML files, it is easier to create, modify and change layouts. But if your tables only require a few parameters to be set, you can create them from the source code as well.

Sample Example

The following example shows you how to create a Table Layout dynamically from the source code

Sample_Table_Activity.java

```
package table.layout2;

import android.app.Activity;
import android.content.Context;
import android.graphics.Color;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

public class Sample_Table_Activity extends Activity {
        /** Called when the activity is first created. */
        @Override
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);
                fillTable();
        }

        private static final String[] firstNames = { "Tom", "Brad", "Jim", "Marlon",
"Keanu", "John", "Bruce" };
        private static final String[] lastNames = { "Hanks", "Pit", "Carrey", "Brando",
"Reeves", "Malkovic", "Willis" };

        public void fillTable() {
                int rowCount = firstNames.length;
                Log.d("Fill Table", "rowCount = " + rowCount);
                TableLayout table = (TableLayout) this.findViewById(R.id.tablelayout);
                for (int i = 0; i < rowCount; i++) {
                        // Below you can choose wich way you want to create your table
                        // Comment on the corresponding part of the code to choose:

                        // Create the table from the source code without xml:
                        /*
                        * TableRow row = new TableRow(this);
                        * row.setBackgroundColor(Color.BLACK); table.addView(row);
                        * fillRow(row, i);
                        */

                        // Create table from xml layout file using layout inflater:
                        fillRow2(table, i);
                }
        }

        public void fillRow(TableRow row, int noRow) {
                // number of rows
                TextView nr = new TextView(this);
                nr.setBackgroundColor(Color.WHITE);
                nr.setTextColor(Color.BLUE);
                nr.setText(String.valueOf(noRow + 1));
                row.addView(nr);
                LinearLayout.LayoutParams llp = (LinearLayout.LayoutParams)
nr.getLayoutParams();
                llp.setMargins(0, 0, 0, 1);
                nr.setLayoutParams(llp);
                nr.setPadding(10, 10, 40, 3);

                // first name
                TextView firstN = new TextView(this);
                firstN.setBackgroundColor(Color.WHITE);
                firstN.setTextColor(Color.BLUE);
                firstN.setText(firstNames[noRow]);
                row.addView(firstN);
                llp = (LinearLayout.LayoutParams) firstN.getLayoutParams();
                llp.setMargins(0, 0, 0, 1);
                firstN.setLayoutParams(llp);
                firstN.setPadding(10, 10, 20, 3);

                // last name
                TextView lastN = new TextView(this);
                lastN.setBackgroundColor(Color.WHITE);
                lastN.setTextColor(Color.BLUE);
                lastN.setText(lastNames[noRow]);
                row.addView(lastN);
                llp = (LinearLayout.LayoutParams) lastN.getLayoutParams();
                llp.setMargins(0, 0, 0, 1);
                lastN.setLayoutParams(llp);
                lastN.setPadding(10, 10, 20, 3);
        }

        public void fillRow2(TableLayout table, int noRow) {
                LayoutInflater inflater = (LayoutInflater)
getSystemService(Context.LAYOUT_INFLATER_SERVICE);
                View fullRow = inflater.inflate(R.layout.row, null, false);
                TextView nr = (TextView) fullRow.findViewById(R.id.nr);
                nr.setText(String.valueOf(noRow + 1));
                TextView fName = (TextView) fullRow.findViewById(R.id.fName);
                fName.setText(firstNames[noRow]);
                fName.setId(noRow + 1);
                TextView lName = (TextView) fullRow.findViewById(R.id.lName);
                lName.setText(lastNames[noRow]);
                table.addView(fullRow);
        }
}
```

main.xml

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/scrollView1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">

                <TableLayout
                android:id="@+id/tablelayout"
```

```
                  android:layout_width="wrap_content"
                  android:layout_height="wrap_content"
                  android:background="@drawable/table_background"
                  android:shrinkColumns="0" >
                  </TableLayout>

         </LinearLayout>

</ScrollView>
```

rows.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableRow xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tableRow1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#000"
     >
         <TextView
                  android:id="@+id/nr"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:background="#FFF"
                  android:textColor="#00F"
                  android:layout_marginBottom="1px"
                  android:paddingLeft="10px"
                  android:paddingTop="10px"
                  android:paddingRight="40px"
                  android:paddingBottom="3px"
                  android:text="nr" />

         <TextView
                  android:id="@+id/fName"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:background="#FFF"
                  android:textColor="#00F"
                  android:layout_marginBottom="1px"
                  android:paddingLeft="10px"
                  android:paddingTop="10px"
                  android:paddingRight="20px"
                  android:paddingBottom="3px"
                  android:text="First Name" />
         <TextView
                  android:id="@+id/lName"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:background="#FFF"
                  android:textColor="#00F"
                  android:layout_marginBottom="1px"
                  android:paddingLeft="10px"
                  android:paddingTop="10px"
                  android:paddingRight="20px"
                  android:paddingBottom="3px"
                  android:text="First Name" />

</TableRow>
```

ref:http://developer.samsung.com/android/technical-docs/Creating-Tables-Using-Table-Layout