**Обзор контейнера TableLayout в Android и пример использования**

Контейнер TableLayout в Android предназначен для организации внутренней иерархии виджетов в виде строк и столбцов. Элемент этого контейнера используется для построения строки в таблице. Каждая строка имеет 0 или больше ячеек и каждая ячейка может содержать один объект View.

Атрибуты контейнера TableLayout

* android:id — уникальный идентификатор, с помощью которого можно работать с контейнером.
* Атрибут android:stretchColumns. С помощью него мы можем растянуть ширину столбца. Индексы столбцов начинаются с нуля и должны быть разделены запятой, например: 1, 2, 5. Также мы можем указать звездочку '\*' в качестве значения, чтобы растянуть все колонки. Пример: мы хотим вставить по одному виджету в трех строках контейнера TableLayout и сделать так, чтобы они занимали всю ширину своей строки. Для этого нам нужно всего лишь добавить атрибут android:stretchColumns со значением «\*».
* android:collapseColumns. С помощью него мы можем сделать столбец невидимым.  Для этого нужно указать один или несколько разделенных запятыми столбцов. Индексы столбцов начинаются с нуля и должны быть разделены запятой, например: 1, 2, 5. Эти столбцы являются частью таблицы, но не видны.
* Атрибут android:layout\_column используется для определения столбца, в который должен быть помещен объект View. По умолчанию любой View помещается в первом неиспользованном столбце.
* Атрибут android:layout\_span используется если объект View должен занимать более одного столбца. Например, если у нас есть строка с тремя элементами и каждый элемент имеет **android:layout\_span=»2″**, то вы будете иметь по крайней мере шесть столбцов в таблице.

Пример использования TableLayout в Android

Для начала создадим [новый Android проект](http://javadevblog.com/pishem-hello-world-na-android-rabotaem-v-android-studio.html). Также Вы можете использовать свой, уже существующий проект: Вам нужно будет просто добавить в него файлы ниже:

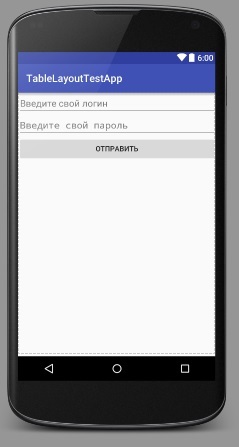
Теперь создадим макет:

Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | <?xml version="1.0" encoding="utf-8"?>  <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:stretchColumns="\*">        <TableRow          android:layout\_width="match\_parent"          android:layout\_height="match\_parent">            <EditText              android:id="@+id/et\_login"              android:hint="Введите свой логин"              android:inputType="text" />      </TableRow>        <TableRow          android:layout\_width="match\_parent"          android:layout\_height="match\_parent">            <EditText              android:id="@+id/et\_password"              android:hint="Введите свой пароль"              android:inputType="textPassword" />      </TableRow>        <TableRow          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content">            <Button              android:id="@+id/btn\_submit"              android:text="Отправить" />      </TableRow>  </TableLayout> |

Как видите, у нас элементарная табличка с полями для ввода логина/пароля и кнопки-подтверждения, которые имитируют простую форму ввода. Для этого нам понадобилось 3 виджета TableRow и корневой TableLayout с указанием растягивать содержимое на всю ширину строки (android:stretchColumns=»\*»).

На превью это будет смотреться так:

[](http://javadevblog.com/wp-content/uploads/2016/09/preview.jpg)

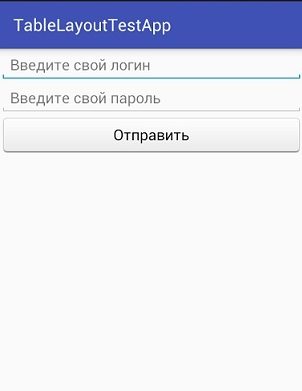
Теперь в классе TableActivity попробуем программно создать все то, что мы сейчас описали в макете:

Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49 | public class TableActivity extends AppCompatActivity {    *@Override*      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);            // при создании объекта View всегда передаем контекст в которм мы его создаем          TableLayout containerTableLayout = new TableLayout(this);          // создаем три растянутые на всю ширину строки          TableRow tableRow1 = new TableRow(this);          TableRow tableRow2 = new TableRow(this);          TableRow tableRow3 = new TableRow(this);            // TableLayout - наследник LinearLayout, поэтому используем настройку          // родительских параметров для ширины и высоты контейнера          containerTableLayout.setLayoutParams(new TableLayout.LayoutParams(                  LinearLayout.LayoutParams.MATCH\_PARENT,                  LinearLayout.LayoutParams.WRAP\_CONTENT          ));          // растягиваем все столбцы с помощью          // программного аналога android:stretchColumns="\*"          containerTableLayout.setStretchAllColumns(true);            // создаем поле для логина          EditText editTextLogin = new EditText(this);          editTextLogin.setHint("Введите свой логин");            // создаем поле для пароля          EditText editTextPassword = new EditText(this);          editTextPassword.setHint("Введите свой пароль");            // создаем кнопку подтверждения          Button buttonSend = new Button(this);          buttonSend.setText("Отправить");            // теперь можем добавить нашпи поля и кнопку          // на свои строки в таблице          tableRow1.addView(editTextLogin);          tableRow2.addView(editTextPassword);          tableRow3.addView(buttonSend);            // добавляем строки в таблицу          containerTableLayout.addView(tableRow1);          containerTableLayout.addView(tableRow2);          containerTableLayout.addView(tableRow3);            setContentView(containerTableLayout);      }  } |

Обратите внимание, что мы могли добавить обработчики событий, идентификаторы для объектов View и многое другое, однако наша задача в этой статье — познакомиться с контейнером TableLayout. Если хотите, попрактикуйтесь сами и добавьте [слушатели на Button](http://javadevblog.com/android-radio-button-primer-ispol-zovaniya-radio-knopki.html) или всплывающие сообщения Toast.

Теперь запустим наш программно генерированный контейнер TableLayout на устройстве и увидим следующее:

[](http://javadevblog.com/wp-content/uploads/2016/09/result.jpg)

Это была которотенькая статья для ознакомления с контейнером TableLayout и его атрибутами. Следите за обновлениями раздела «[Разработка под Android](http://javadevblog.com/category/android)«.