Toolbar альтернатива ActionBar

Toolbar — виджет, который позволяет нам сохранить функциональность крайне удобного во многих приложениях ActionBar-а, и, в то же самое время, дает возможность очень легко и просто стилизовать этот элемент. Toolbar, как вы увидите дальше, является частью Layout-а, что и облегчает доступ к его атрибутам. Рассмотрим как можно создать этот виджет.

1. Добавим необходимые зависимости для сборки приложения:

|  |  |
| --- | --- |
| 1  2  3  4 | dependencies {      compile fileTree(dir: 'libs', include: ['\*.jar'])      compile 'com.android.support:appcompat-v7:21.0.3'  } |

Кстати, использование библиотеки **appcompat** позволяет нам задать диапазон версий SDK, начиная от старых и заканчивая максимально новыми:

|  |  |
| --- | --- |
| 1  2 | minSdkVersion 9  targetSdkVersion 21 |

2. Переопределим одну из тем с поддержкой Action Bar-a, добавив атрибуты, скрывающие сам бар —**res/values/styles.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <resources>      <style name="appTheme" parent="Theme.AppCompat.Light.DarkActionBar">          <item name="android:windowNoTitle">true</item>          <item name="windowActionBar">false</item>      </style>  </resources> |

В файле манифеста необходимо указать для тега application эту тему:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android"      package="ru.androiddocs.toolbarexample" >        <application          android:allowBackup="true"          android:icon="@drawable/ic\_launcher"          android:label="@string/app\_name"          android:theme="@style/appTheme" >          <activity              android:name=".MainActivity"              android:label="@string/app\_name" >              <intent-filter>                  <action android:name="android.intent.action.MAIN"/>                  <category android:name="android.intent.category.LAUNCHER"/>              </intent-filter>          </activity>      </application>    </manifest> |

3. Добавим виджет в верстку нашего экрана — **res/layout/activity\_main.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"      xmlns:tools="http://schemas.android.com/tools"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      tools:context=".MainActivity">        <android.support.v7.widget.Toolbar          xmlns:android="http://schemas.android.com/apk/res/android"          android:id="@+id/toolbar"          android:layout\_width="match\_parent"          android:layout\_height="50dp"          android:background="#ff15cc48" />        <TextView          android:text="@string/hello\_world"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_below="@+id/toolbar" />    </RelativeLayout> |

Как видите, мы добавили напрямую в Layout виджет **Toolbar** (из Support Library), добавив для его отображения несколько свойств. В частности, обратите внимание на атрибут **layout\_height** со значением 50dp — это стандартная высота Action Bar-a. Также мы задали цвет фона — светло-зеленый.

4. Строковые ресурсы **res/values/strings.xml**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <?xml version="1.0" encoding="utf-8"?>  <resources>      <string name="app\_name">Toolbar Example</string>      <string name="hello\_world">Hello world!</string>      <string name="action\_settings">Settings</string>  </resources> |

5. Описание меню опций res/menu/menu\_main.xml

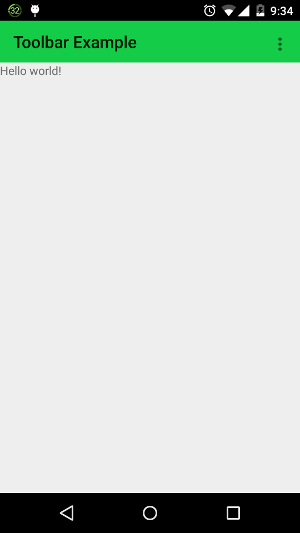
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <menu xmlns:android="http://schemas.android.com/apk/res/android"      xmlns:app="http://schemas.android.com/apk/res-auto"      xmlns:tools="http://schemas.android.com/tools"      tools:context=".MainActivity">        <item android:id="@+id/action\_settings"          android:title="@string/action\_settings"          android:orderInCategory="100"          app:showAsAction="never" />    </menu> |

6. Добавим класс Активити с методами для создания меню опций (на данный момент нам важно увидеть, что мы можем заменить Тулбаром Action Bar, сохранив его функциональность).

|  |  |
| --- | --- |
| 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 | package ru.androiddocs.toolbarexample;    import android.support.v7.app.ActionBarActivity;  import android.os.Bundle;  import android.support.v7.widget.Toolbar;  import android.view.Menu;  import android.view.MenuItem;      public class MainActivity extends ActionBarActivity {        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);          setSupportActionBar(toolbar);      }        @Override      public boolean onCreateOptionsMenu(Menu menu) {          getMenuInflater().inflate(R.menu.menu\_main, menu);          return true;      }        @Override      public boolean onOptionsItemSelected(MenuItem item) {            int id = item.getItemId();          if (id == R.id.action\_settings) {              return true;          }          return super.onOptionsItemSelected(item);      }  } |

Чтобы наш Toolbar отобразился в методе onCreate() необходимо вызвать метод **setSupportActionBar()**, передав ему ссылку на виджет.

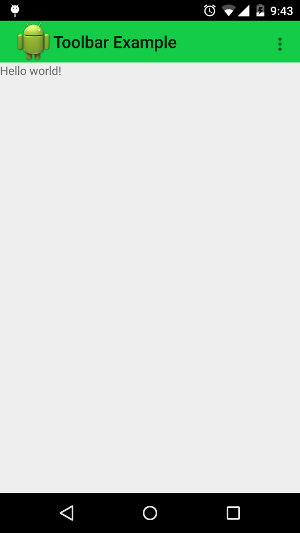
7. Запустим приложение и увидим стандартный Action Bar:



**Как добавить иконку приложения в Toolbar**

Добавить иконку в Toolbar можно с помощью метода **setLogo()**, передав в качестве параметра ссылку на ресурс:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | @Override  protected void onCreate(Bundle savedInstanceState) {      super.onCreate(savedInstanceState);      setContentView(R.layout.activity\_main);        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);      setSupportActionBar(toolbar);        toolbar.setLogo(R.drawable.ic\_launcher);  } |



**Добавляем заголовок и подзаголовок**

При добавлении заголовка и подзаголовка иконку лучше не отображать. Используем методы **setTitle()**и **setSubtitle()**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);          setSupportActionBar(toolbar);            toolbar.setTitle("title");          toolbar.setSubtitle("Sub title");      } |

Также мы чуток изменим высоту нашего Тулбара, чтобы подзаголовок целиком поместился:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <android.support.v7.widget.Toolbar      xmlns:android="http://schemas.android.com/apk/res/android"      android:id="@+id/toolbar"      android:layout\_width="match\_parent"      android:layout\_height="60dp"      android:background="#ff15cc48" /> |

