Пользовательский диалог в Android

Рассмотрим создание пользовательского диалога на платформе Android. Будут показаны два очень похожих способа создания диалога.

Итак, приступим. Для начала отредактируем файл разметки (в примере этот файл имеет имя mail.xml):

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/rootView"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="CustomDialog" />

<Button

android:id="@+id/button\_dialog1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Dialog (Method 1)" >

</Button>

<Button

android:id="@+id/button\_dialog2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Dialog (Method 2)" >

</Button>

</LinearLayout>

Каждая из двух кнопок будут отвечать за свой способ отображения диалога.

Ниже приведены два класса для каждого способа отображения диалога:

CustomDialog1.java

CustomDialog2.java

Различия этих двух классов в методе init(). В первом случае диалог создается с использованием конструктора Dialog():

dialog = new Dialog(activity);

А разметка присваивается с помощью метода setContentView() класса Dialog:

dialog.setContentView(R.layout.dialog);

Во втором случае диалог создается с использованием класса AlertDialog.Builder():

AlertDialog.Builder builder = new AlertDialog.Builder( this.activity );

*//...*

dialog = builder.create();

А разметка создается отдельно с помощью LayoutInflater:

LayoutInflater inflater = LayoutInflater.from( this.activity );

View layout = inflater.inflate(R.layout.dialog, null);

И присваивается объекту builder, ответственному за создания диалога:

builder.setView(layout);

В обеих способах используется одна и та же разметка для диалога (файл /res/layout/dialog.xml):

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_gravity="center" >

<EditText

android:id="@+id/dialog\_edit\_box"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Enter text..." />

<TextView

android:id="@+id/dialog\_info"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="..." />

<Button

android:id="@+id/dialog\_button"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Click me!" />

</LinearLayout>

Класс Activity:

package com.seostella.customdialog;

import android.app.Activity;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

public class CustomDialogActivity extends Activity {

private CustomDialog1 customDialog1;

private CustomDialog2 customDialog2;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

customDialog1 = new CustomDialog1( this );

Button buttonDialog1 = (Button) findViewById(R.id.button\_dialog1);

buttonDialog1.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

customDialog1.show();

}

});

customDialog2 = new CustomDialog2(this);

Button buttonDialog2 = (Button) findViewById(R.id.button\_dialog2);

buttonDialog2.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

customDialog2.show();

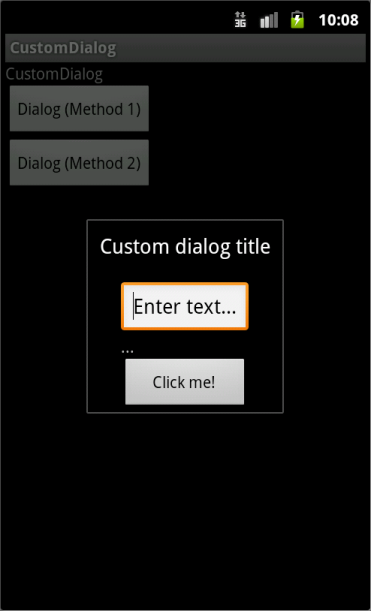
}

});

}

}

При нажатии на первую кнопку отображается следующий диалог:



При нажатии на вторую кнопку отображается следующий диалог:

