**Пример работы с DatePickerDialog и TimePickerDialog в Android**

[09/05/2016](http://prologistic.com.ua/primer-raboty-s-datepickerdialog-i-timepickerdialog-v-android.html)  [prologistic](http://prologistic.com.ua/author/prologistic)  [Android](http://prologistic.com.ua/category/android)

В этом уроке мы создадим простое приложение с Date Picker и Time Picker Dialog. Эти компоненты используются для выбора даты и времени непосредственно самим пользователем. В коде они реализованы с помощью классов DatePickerDialog, TimePickerDialog и класса Calendar. Подробнее читайте ниже в статье.

**Краткий обзор DatePickerDialog и TimePickerDialog в Android**

Обычно выбор даты и времени в андроид реализуется с помощью независимых виджетов, но по отдельности они неудобны и занимают больше места на экране, поэтому их чаще используют внутри [Dialog](http://prologistic.com.ua/primer-alertdialog-dialogovoe-okno-v-android.html" \t "_blank). Разработчики системы Android это заметили и создали удобные классы DatePickerDialog и TimePickerDialog соответственно для выбора даты и времени.

Для взаимодействия с пользователем в каждом из этих классов используются интерфейсы обратного вызоваonDateSetListener() и onTimeSetListener().

В конструкторе класса DatePickerDialog целых 5 параметров:

1. Context — контекст приложения
2. Функция обратного вызова: onDateSet() вызывается, когда пользователь устанавливает дату со следующими параметрами:
   * int year — получает выбранный год из диалога
   * int monthOfYear — хранит выбранный месяц из диалога
   * int dayOfMonth — хранит выбранный день из диалога
3. int mYear — год, который отображается при создании диалога
4. int mMonth — месяц
5. int mDay — день

В конструкторе класса TimePickerDialog также 5 параметров:

1. Context — требуется контекст приложения
2. Функция обратного вызова: onTimeSet() вызывается, когда пользователь устанавливает время со следующими параметрами:
   * int hourOfDay — хранит выбранный час дня из диалога
   * int minute — хранит выбранную минуту из диалога
3. int mHours — час, который отображается при создании диалога
4. int mMinute — минута, которая отображается при создании диалога
5. boolean mBool — если false, то формат времени 24 часа, если true — 12 часов.

Подробнее о диалогах в Android читайте в [этой статье](http://prologistic.com.ua/primer-alertdialog-dialogovoe-okno-v-android.html) и [по этой ссылке](http://prologistic.com.ua/dialogi-v-android-primer-raboty-s-progressdialog.html).

**Практика. Создание DatePickerDialog и TimePickerDialog в Android**

В макете **activity\_main.xml** будет 2 виджета [Button](http://prologistic.com.ua/sozdaem-knopku-v-android-primer-raboty-s-vidzhetom-button-v-android.html" \t "_blank): для вызова DatePickerDialogи TimePickerDialog. Выбранные пользователем дата и время будут сохранены в соответствующие поля EditText:

Java

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <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"      android:paddingBottom="@dimen/activity\_vertical\_margin"      android:paddingLeft="@dimen/activity\_horizontal\_margin"      android:paddingRight="@dimen/activity\_horizontal\_margin"      android:paddingTop="@dimen/activity\_vertical\_margin"      tools:context="ua.com.prologistic.datetimepickerdialogsproject.MainActivity">        <EditText          android:id="@+id/picked\_date"          android:hint="@string/date\_edit\_text\_hint"          android:layout\_width="200dp"          android:layout\_height="wrap\_content"          android:layout\_marginTop="85dp"          android:layout\_alignParentTop="true"          android:layout\_toLeftOf="@+id/btn\_date"          android:layout\_alignParentLeft="true"          android:layout\_alignParentStart="true"          android:inputType="date" />        <Button          android:id="@+id/btn\_date"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:text="@string/btn\_date\_text"          android:layout\_above="@+id/picked\_time"          android:layout\_alignParentRight="true"          android:layout\_alignParentEnd="true" />        <EditText          android:id="@+id/picked\_time"          android:layout\_width="200dp"          android:layout\_height="wrap\_content"          android:hint="@string/time\_edit\_text\_hint"          android:layout\_alignParentLeft="true"          android:layout\_alignParentStart="true"          android:layout\_below="@+id/picked\_date"          android:layout\_toLeftOf="@+id/btn\_date"          android:layout\_toStartOf="@+id/btn\_date"          android:inputType="time"/>        <Button          android:id="@+id/btn\_time"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_alignLeft="@+id/btn\_date"          android:layout\_alignStart="@+id/btn\_date"          android:layout\_below="@+id/btn\_date"          android:text="@string/btn\_time\_text" />    </RelativeLayout> |

Строки были вынесены в файл **strings.xml**:

Java

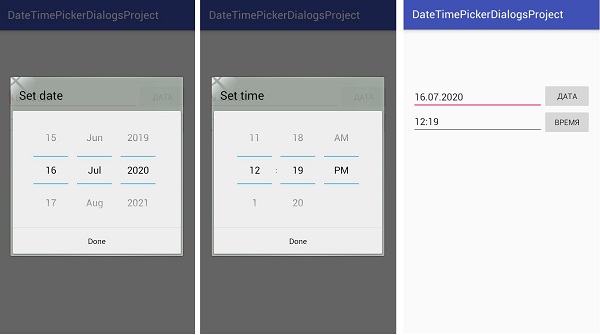
|  |
| --- |
| <resources>      <string name="app\_name">DateTimePickerDialogsProject</string>      <string name="time\_edit\_text\_hint">здесь будет время..</string>      <string name="date\_edit\_text\_hint">здесь будет дата..</string>      <string name="btn\_time\_text">Время</string>      <string name="btn\_date\_text">Дата</string>  </resources> |

В классе MainActivity нужно инициализировать слушатель и вызывать нужный метод в зависимости от нажатой кнопки:

Java

|  |
| --- |
| package ua.com.prologistic.datetimepickerdialogsproject;    import android.app.DatePickerDialog;  import android.app.TimePickerDialog;  import android.os.Bundle;  import android.support.v7.app.AppCompatActivity;  import android.view.View;  import android.widget.Button;  import android.widget.DatePicker;  import android.widget.EditText;  import android.widget.TimePicker;    import java.util.Calendar;    public class MainActivity extends AppCompatActivity implements          View.OnClickListener {        private Button btnDatePicker, btnTimePicker;      private EditText editTextDate, editTextTime;        // делаем переменные даты/времени полями, т.к. в реальных      // приложениях они чаще всего используются и в других местах.      private int mYear, mMonth, mDay, mHour, mMinute;    *@Override*      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            btnDatePicker = (Button) findViewById(R.id.btn\_date);          btnTimePicker = (Button) findViewById(R.id.btn\_time);          editTextDate = (EditText) findViewById(R.id.picked\_date);          editTextTime = (EditText) findViewById(R.id.picked\_time);            btnDatePicker.setOnClickListener(this);          btnTimePicker.setOnClickListener(this);        }    *@Override*      public void onClick(View v) {          int id = v.getId();            switch (id) {              case R.id.btn\_date:                  // вызываем диалог с выбором даты                  callDatePicker();                  break;                case R.id.btn\_time:                  // вызываем диалог с выбором времени                  callTimePicker();                  break;          }      }        private void callTimePicker() {          // получаем текущее время          final Calendar cal = Calendar.getInstance();          mHour = cal.get(Calendar.HOUR\_OF\_DAY);          mMinute = cal.get(Calendar.MINUTE);            // инициализируем диалог выбора времени текущими значениями          TimePickerDialog timePickerDialog = new TimePickerDialog(this,                  new TimePickerDialog.OnTimeSetListener() {  *@Override*                      public void onTimeSet(TimePicker view, int hourOfDay, int minute) {                          String editTextTimeParam = hourOfDay + " : " + minute;                          editTextTime.setText(editTextTimeParam);                      }                  }, mHour, mMinute, false);          timePickerDialog.show();      }        private void callDatePicker() {          // получаем текущую дату          final Calendar cal = Calendar.getInstance();          mYear = cal.get(Calendar.YEAR);          mMonth = cal.get(Calendar.MONTH);          mDay = cal.get(Calendar.DAY\_OF\_MONTH);            // инициализируем диалог выбора даты текущими значениями          DatePickerDialog datePickerDialog = new DatePickerDialog(this,                  new DatePickerDialog.OnDateSetListener() {  *@Override*                      public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {                          String editTextDateParam = dayOfMonth + "." + (monthOfYear + 1) + "." + year;                          editTextDate.setText(editTextDateParam);                      }                  }, mYear, mMonth, mDay);          datePickerDialog.show();      }  } |

Результат выполнения программы для работы с DatePickerDialog и TimePickerDialog в Android:

[](http://prologistic.com.ua/wp-content/uploads/2016/05/datepickerdialog.jpg)