Настройки в Android-приложениях

Настройки являются важной частью приложений на Android *(и не только на Android — здесь и далее примечание переводчика)*. Это очень важно — позволять пользователям изменять настройки приложения, в зависимости от их предпочтений.  
  
Существует два пути работы с настройками в Android — можно создать файл *preferences.xml* в директории *res/xml*, либо работать с настройками из кода. В данной статье я покажу как работать с настройками, используя *preferences.xml* файл.  
  
  
  
Элементы настроек имеют следующие атрибуты:

* *android:key* — имя настройки, по поторому в дальнейшем можно получить ее значение
* *android:title* — заголовок элемента настройки
* *android:summary* — краткое описание элемента настройки
* *android:defaultValue* — значение по умолчанию

В настоящее время доступны следующие типы элементов настроек:

* *CheckBoxPreference* — простой чекбокс, который возвращает значения true или false.
* *ListPreference* — группа переключателей (radioGroup), из которых может быть выбран только один элемент. Атрибут *android:entries* указывает на массив со значениями в *res/values/arrays.xml*, а *android:entryValues* на массив с подписями.
* *EditTextPreference* — показывает диалоговое окно с полем ввода. Возвращает строку в качестве значения.
* *RingtonePreference* — группа переключателей с выбором рингтона.
* *Preference* — настройка, работающая как кнопка.
* *PreferenceScreen* — экран с настройками. Когда один PreferenceScreen вложен в другой, то открывается новый экран с настройками.
* *PreferenceCategory* — категория настроек.

|  |  |
| --- | --- |
| *Экран с настройками* https://habrastorage.org/storage/a37f9b3f/1a7c8e9f/1677bdc3/cf4aec48.png | *EditTextPreference* https://habrastorage.org/storage/42404c69/35d08308/efa19ca5/05fc3013.png |
| *ListPreference* https://habrastorage.org/storage/9d6616b5/63afca14/e5048eef/4455dd76.png | *RingtonePreference* https://habrastorage.org/storage/10649e32/c98c2ab4/253c7239/603b9f7e.png |
| *PreferenceScreen* https://habrastorage.org/storage/c2831688/f8d94d53/63dcbe89/8698e6a8.png |  |

Скриншоты выше были сгенерированы при помощи следующего *preferences.xml*:

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

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

<PreferenceCategory

android:title="First Category">

<CheckBoxPreference

android:title="Checkbox Preference"

android:defaultValue="false"

android:summary="This preference can be true or false"

android:key="checkboxPref" />

<ListPreference

android:title="List Preference"

android:summary="This preference allows to select an item in a array"

android:key="listPref"

android:defaultValue="digiGreen"

android:entries="@array/listArray"

android:entryValues="@array/listValues" />

</PreferenceCategory>

<PreferenceCategory

android:title="Second Category">

<EditTextPreference

android:name="EditText Preference"

android:summary="This allows you to enter a string"

android:defaultValue="Nothing"

android:title="Edit This Text"

android:key="editTextPref" />

<RingtonePreference

android:name="Ringtone Preference"

android:summary="Select a ringtone"

android:title="Ringtones"

android:key="ringtonePref" />

<PreferenceScreen

android:key="SecondPrefScreen"

android:title="Second PreferenceScreen"

android:summary="This is a second PreferenceScreen">

<EditTextPreference

android:name="An other EditText Preference"

android:summary="This is a preference in the second PreferenceScreen"

android:title="Edit text"

android:key="SecondEditTextPref" />

</PreferenceScreen>

<Preference

android:title="Custom Preference"

android:summary="This works almost like a button"

android:key="customPref" />

</PreferenceCategory>

</PreferenceScreen>

Атрибуты *android:entries* и *android:entryValues* у *ListPreference* ссылаются на *@array/listArray* и *@array/listValues* соответственно. Значения берутся из *res/values/arrays.xml*, который в нашем случае выглядит следующим образом:

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

<resources>

<string-array name="listArray">

<item>Number 1</item>

<item>Number 2</item>

<item>Number 3</item>

<item>Number 4</item>

</string-array>

<string-array name="listValues">

<item>1</item>

<item>2</item>

<item>3</item>

<item>4</item>

</string-array>

</resources>

Для того, чтобы показать пользователю экран с настройками, небходимо создать активити, унаследованное от PreferenceActivity. Пример активити:

package org.kaloer.preferenceexample;

import android.app.Activity;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.preference.Preference;

import android.preference.PreferenceActivity;

import android.preference.Preference.OnPreferenceClickListener;

import android.widget.Toast;

public class Preferences extends PreferenceActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

addPreferencesFromResource(R.xml.preferences);

// Get the custom preference

Preference customPref = (Preference) findPreference("customPref");

customPref.setOnPreferenceClickListener(new OnPreferenceClickListener() {

public boolean onPreferenceClick(Preference preference) {

Toast.makeText(getBaseContext(),

"The custom preference has been clicked",

Toast.LENGTH\_LONG).show();

SharedPreferences customSharedPreference = getSharedPreferences(

"myCustomSharedPrefs", Activity.MODE\_PRIVATE);

SharedPreferences.Editor editor = customSharedPreference

.edit();

editor.putString("myCustomPref",

"The preference has been clicked");

editor.commit();

return true;

}

});

}

}

А вызвать активити с настройками можно, нажав на кнопку на нашем главном активити:

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

Button prefBtn = (Button) findViewById(R.id.prefButton);

prefBtn.setOnClickListener(new OnClickListener() {

public void onClick(View v) {

Intent settingsActivity = new Intent(getBaseContext(),

Preferences.class);

startActivity(settingsActivity);

}

});

}

Для того, чтобы использовать выставленными в настройках значениями, добавим метод *getPrefs()* в главное активити, который нужно вызывать на событии *onStart()*. Нужно использовать именно *onStart()*, а не *onCreate()*, для того, чтобы быть уверенным в том, что используются актуальные настройки, а не те, что были во время создания гланого активити. Наш метод *getPrefs()* может выглядеть примерно вот так:

boolean CheckboxPreference;

String ListPreference;

String editTextPreference;

String ringtonePreference;

String secondEditTextPreference;

String customPref;

private void getPrefs() {

// Get the xml/preferences.xml preferences

SharedPreferences prefs = PreferenceManager

.getDefaultSharedPreferences(getBaseContext());

CheckboxPreference = prefs.getBoolean("checkboxPref", true);

ListPreference = prefs.getString("listPref", "nr1");

editTextPreference = prefs.getString("editTextPref",

"Nothing has been entered");

ringtonePreference = prefs.getString("ringtonePref",

"DEFAULT\_RINGTONE\_URI");

secondEditTextPreference = prefs.getString("SecondEditTextPref",

"Nothing has been entered");

// Get the custom preference

SharedPreferences mySharedPreferences = getSharedPreferences(

"myCustomSharedPrefs", Activity.MODE\_PRIVATE);

customPref = mySharedPreferences.getString("myCusomPref", "");

}

И на последок, не забудте добавить созданное активити с настройками в *androidmanifest.xml* и добавить новую строку с именем «set\_preferences», для обозначения заголовка экрана с настройками, например «Preferences».

<activity

android:name=".Preferences"

android:label="@string/set\_preferences">

</activity>

В итоге, мы получим следующее:

|  |  |
| --- | --- |
| *Экран с главным активити* https://habrastorage.org/storage/2c224db9/a0316291/e0c6b49a/6814ef9c.png | *Активити с настройками* https://habrastorage.org/storage/ceb0e6dd/ef3a1301/f758542d/9e5ef417.png |