Spinner

[Общая информация](http://developer.alexanderklimov.ru/android/views/spinner.php#common)  
[Быстрый старт](http://developer.alexanderklimov.ru/android/views/spinner.php#quickstart)  
[Используем адаптер](http://developer.alexanderklimov.ru/android/views/spinner.php#adapter)  
[За честные выборы! - что выбрал пользователь](http://developer.alexanderklimov.ru/android/views/spinner.php#select)  
[Тонкая настройка](http://developer.alexanderklimov.ru/android/views/spinner.php#custom)  
[Не выбирать элемент при запуске](http://developer.alexanderklimov.ru/android/views/spinner.php#emptyitem)  
[Забудьте про setPrompt()](http://developer.alexanderklimov.ru/android/views/spinner.php#setprompt)  
[Режим android:spinnerMode="dialog"](http://developer.alexanderklimov.ru/android/views/spinner.php#spinnermode)

Общая информация

Компонент **Spinner** похож на выпадающий список (ComboBox), используемый в OC Windows. В закрытом состоянии компонент показывает одну строчку, при раскрытии выводит список в виде диалогового окна с переключателями.

Находится в разделе **Widgets**.

Быстрый старт

Сначала покажу быстрый способ использования элемента. При добавлении элемента на экран отображается просто полоска со строкой **Item1**. В основном настройка происходит программным путем. Но можно и через XML. Добавим в строковый файл ресурсов **strings.xml** несколько элементов массива:

<string-array name="catlist">

<item>Барсик</item>

<item>Мурзик</item>

<item>Васька</item>

<item>Рыжик</item>

</string-array>

Теперь осталось в атрибуте **android:entries** указать на созданный массив и компонент **Spinner** будет заполнен данными. Запустите проект и проверьте.

<Spinner

android:id="@+id/spinner"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:entries="@array/catlist" />

Если нужно из программы узнать, какой пункт из выпадающего списка выбран в **Spinner**, то можно использовать такой код, например, при нажатии кнопки:

Spinner spinner = (Spinner) findViewById(R.id.spinner);

String selected = spinner.getSelectedItem().toString();

Toast.makeText(getApplicationContext(), selected, Toast.LENGTH\_SHORT).show();

Если нужен не текст, а номер позиции, то вызывайте метод **getSelectedItemPosition()**

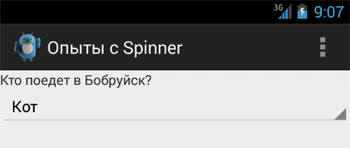
Если вам нужно получить выбранный элемент сразу в момент выбора, то используйте метод **setOnItemSelectedListener()**, который описан [ниже](http://developer.alexanderklimov.ru/android/views/spinner.php#select).

Используем адаптер

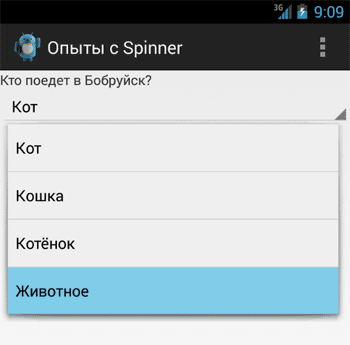
Как и в случае с компонентом **ListView**, **Spinner** использует адаптер данных для связывания содержимого из набора данных с каждым пунктом в списке. Для загрузки данных нужно:

* Получить экземпляр компонента **Spinner**
* Настроить адаптер данных для связывания
* Вызвать метод **setAdapter()**

В закрытом состоянии



В раскрытом состоянии



Данные в закрытом и раскрытом состоянии **Spinner** отображает по разному. Поэтому необходимо создавать макеты шаблонов для обоих состояний. Android предоставляет несколько своих собственных ресурсов для **Spinner** для простых задач. Например, есть ресурс **android.R.layout.simple\_spinner\_item** для создания представления для каждого элемента списка. Ресурс **android.R.layout.simple\_spinner\_dropdown\_item** служит шаблоном для раскрывающего списка.

Создадим строковый массив в файле **strings.xml**:

<string-array name="animals">

<item>Кот</item>

<item>Кошка</item>

<item>Котёнок</item>

<item>Животное</item>

</string-array>

Загрузим строковый массив с именем **animals** в экземпляр класса **ArrayAdapter** при помощи метода **createFromResource()**:

// Получаем экземпляр элемента Spinner

final Spinner spinner = (Spinner)findViewById(R.id.spinner);

// Настраиваем адаптер

ArrayAdapter<?> adapter =

ArrayAdapter.createFromResource(this, R.array.animals, android.R.layout.simple\_spinner\_item);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

// Вызываем адаптер

spinner.setAdapter(adapter);

Запустив программу, вы увидите работающий пример, как на картинках, представленных выше.

По умолчанию выводится первый элемент списка. С помощью метода **setSelection()** можно установить нужный элемент по умолчанию, указав индекс из строкового ресурса.

spinner.setSelection(2);

За честные выборы! - что выбрал пользователь

Нам интересно узнать, что именно выбрал пользователь из списка и обработать эту информацию.

Хотя в приведенном примере у нас нет выбора, так как любой другой вариант кроме последнего приведет к уменьшению вашей кармы, вы в своих программах будете использовать более гуманные варианты списка.

Нам нужно получить выбранный пользователем пункт в компоненте **Spinner** при помощи метода **setOnItemSelectedListener()** и реализовать метод **onItemSelected()** класса**AdapterView.OnItemSelectedListener**:

spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

public void onItemSelected(AdapterView<?> parent,

View itemSelected, int selectedItemPosition, long selectedId) {

String[] choose = getResources().getStringArray(R.array.animals);

Toast toast = Toast.makeText(getApplicationContext(),

"Ваш выбор: " + choose[selectedItemPosition], Toast.LENGTH\_SHORT);

toast.show();

}

public void onNothingSelected(AdapterView<?> parent) {

}

});

Теперь при выборе любого пункта вы получите всплывающее сообщение о выбранном пункте. Обратите внимание, что нам также пришлось реализовать вызов обратного вызова**onNothingSelected()**.

В начале статьи показывался более простой способ с использованием метода **getSelectedItem()**, который достаточен для большинства случаев.

Предупредить компонент об изменении пунктов

Если в приложении вы изменили состав выпадающего списка, то необходимо сообщить компоненту **Spinner**, чтобы он показывал обновленный список. Сделать это можно при помощи метода адаптера **notifyDataSetChanged()**.

adapter.notifyDataSetChanged();

Найти позицию по слову

Если мы хотим узнать, в какой позиции находится то или иное слово, то нужно получить адапетр через метод **getAdapter()**, а затем уже и позицию.

String kitten = "Котёнок"; // нужная строка

Spinner spinner = (Spinner) findViewById(R.id.spinner);

ArrayAdapter adapter = (ArrayAdapter) spinner.getAdapter();

int position = adapter.getPosition(kitten);

Toast.makeText(getApplicationContext(), "Котёнок содержится в позиции " + position,

Toast.LENGTH\_SHORT).show();

Тонкая настройка - своя разметка для Spinner

Вы можете установить собственный фон, но не можете установить, к примеру, цвет и размер текста в настройках свойств. В предыдущих примерах мы видели, что при подключении к адаптеру используются системные разметки **android.R.layout.simple\_spinner\_item** и **android.R.layout.simple\_spinner\_dropdown\_item**. Ничто не мешает вам посмотреть исходники данных файлов и создать файлы для собственной разметки, которые потом можно подключить к адаптеру.

Давайте создадим собственную разметку с значками. В папке **res/layout** создаём файл **row.xml**:

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

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

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal" >

<ImageView

android:id="@+id/icon"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/ic\_launcher" />

<TextView

android:id="@+id/weekofday"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

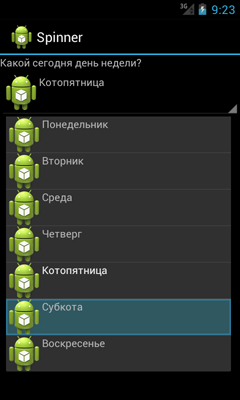
</LinearLayout>

Осталось в коде заменить две строки на одну:

// Подключаем свой шаблон с значками

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,

R.layout.row, R.id.weekofday, DayOfWeek);



В примере использовался один общий файл, но можете создать два отдельных шаблона для закрытого и раскрытого вида элемента. Например, так (простейший вариант):

spinner.xml

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

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

android:id="@+id/spinnertext"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="40sp"

android:textColor="@color/buttontext" />

spinner\_dropdown\_item.xml

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

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

android:id="@+id/spinnerdropdown"

android:singleLine="true"

android:layout\_width="fill\_parent"

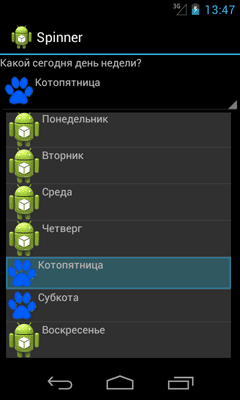
android:layout\_height="wrap\_content"

android:textColor="@color/buttontext"

android:textSize="40sp" />

В принципе, вы можете установить свой значок для каждого пункта, вам нужно создать свой адаптер под свои нужды. Создадим новый класс на основе **ArrayAdapter** и реализуем задачу. Теперь у двух любимых дней недели будет выводиться лапочка.

Показать код (щелкните мышкой)



Программная настройка цвета и размера текста для первой строчки

В сети нашёл пример программной установки цвета и размера текста для первой строчки элемента в закрытом состоянии. Может кому пригодится.

spinner.setOnItemSelectedListener(new OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view, int pos,

long id) {

// TODO Auto-generated method stub

((TextView) parent.getChildAt(0)).setTextColor(Color.MAGENTA);

((TextView) parent.getChildAt(0)).setTextSize(12);

}

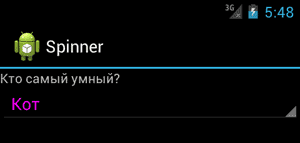
@Override

public void onNothingSelected(AdapterView<?> arg0) {

// TODO Auto-generated method stub

}

});



Не выбирать элемент при запуске

Иногда хочется, что при запуске не выбирался первый элемент списка, как это происходит по умолчанию. Решение в лоб - добавить первым элементом пустую строку или текст "Выберите..." не слишком красив, так как при раскрытии списка мы увидим эти элементы, которые только портят картину. В сети нашёл вариант, использующий собственный адаптер.

private String[] cats = { "Васька", "Мурзик", "Барсик", "Рыжик" };

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_test);

Spinner spinner = (Spinner) findViewById(R.id.spinner);

CustomAdapter adapter = new CustomAdapter(this,

android.R.layout.simple\_spinner\_item, cats);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinner.setAdapter(adapter);

spinner.setOnItemSelectedListener(new OnItemSelectedListener() {

@Override

public void onNothingSelected(AdapterView<?> parent) {

}

@Override

public void onItemSelected(AdapterView<?> parent, View view,

int pos, long id) {

// Set adapter flag that something has been chosen

CustomAdapter.flag = true;

}

});

}

CustomAdapter.java

package ru.alexanderklimov.fortest;

import android.content.Context;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ArrayAdapter;

import android.widget.TextView;

public class CustomAdapter extends ArrayAdapter {

private Context context;

private int textViewResourceId;

private String[] objects;

public static boolean flag = false;

public CustomAdapter(Context context, int textViewResourceId,

String[] objects) {

super(context, textViewResourceId, objects);

this.context = context;

this.textViewResourceId = textViewResourceId;

this.objects = objects;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

if (convertView == null)

convertView = View.inflate(context, textViewResourceId, null);

if (flag != false) {

TextView tv = (TextView) convertView;

tv.setText(objects[position]);

}

return convertView;

}

}

Попробуйте этот вариант, может он подойдёт вам.

Забудьте про setPrompt()

В старых версиях Android 2 для компонента **Spinner** можно было вывести заголовок с помощью методов **setPrompt()** или **setPromptId()**. В Android 4 и выше заголовки не выводятся, поэтому смысла использовать эти методы для новых проектов нет, хотя они и не приводят к ошибке. Используйте, если вам важна совместимость.

Режим android:spinnerMode="dialog"

У компонента есть "секретный" атрибут **android:spinnerMode**, у которого можно установить значение **dialog**. В этом случае при раскрытии списка задняя активность затемняется. Это хорошо заметно на белом фоне. Проверьте самостоятельно.

<Spinner

android:id="@+id/spinner"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:spinnerMode="dialog" />

Дополнительное чтение

[Настройка Spinner](http://developer.alexanderklimov.ru/android/design/spinner-style.php)

[Продвинутые примеры с Spinner](http://developer.alexanderklimov.ru/android/views/spinner-advanced.php)

Центрирование текста: [set view text align at center in spinner in android - Stack Overflow](http://stackoverflow.com/questions/7511049/set-view-text-align-at-center-in-spinner-in-android)