

Google Android

... ЭТО НЕСЛОЖНО



Поиск по сайту

Home

▼ Уроки

▼ Статьи

▼ Новости

▼ About

▼ Партнеры

Форум

Яндекс.Директ

[разработка приложений](#)

18+

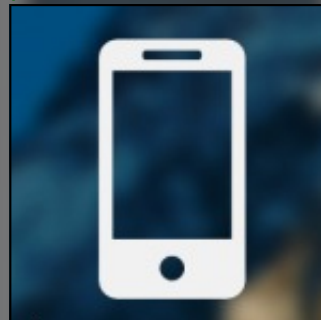
appsmob.ru


[Разработка приложений на android](#)

fixapp.ru


[Заказать мобильное приложение](#)

spacewell.ru



LANGUAGE



Присоединяйтесь к
Jabber чату
StartAndroid

ВАКАНСИИ РАЗРАБОТЧИКС

На нашем форуме
рекрутеры
периодически
пополняют [список
вакансий](#) Android
разработчиков.
Будет время -
загляните,
возможно вас
заинтересуют эти
предложения.

Урок 46. События ExpandableListView

МАТЕРИАЛЫ ПО СМЕЖНЫМ ТЕМАМ

- [Урок 14. Меню, группы, порядок. MenuInflater и xml-меню.](#)
- [Урок 47. Обзор адаптеров](#)
- [Урок 41. Используем LayoutInflater для создания списка](#)
- [Урок 42. Список - ListView](#)
- [Урок 43. Одиночный и множественный выбор в ListView](#)
- [Урок 44. События в ListView](#)
- [Урок 45. Список-дерево ExpandableListView](#)
- [Урок 51. SimpleAdapter, добавление и удаление записей](#)
- [Урок 52. SimpleCursorAdapter, пример использования](#)
- [Урок 62. Диалоги. AlertDialog. Список](#)
- [Урок 63. Диалоги. AlertDialog. Список с одиночным выбором](#)
- [Урок 108. Android 3. ActionBar. Навигация - табы и выпадающий список](#)



Создано 29.12.2011 08:00



Автор: damager82

В этом уроке:

- обрабатываем события дерева-списка

ПИАР-ПОДДЕРЖКА

Вы создали приложение/проект/стартап, о котором хотите рассказать? У меня к вам есть [предложение](#).

ПОДДЕРЖКА ПРОЕКТА

[Яндекс](#)

410011180491924

Alfa-Bank

5486734918678877

[WebMoney](#)

R248743991365

Z551306702056

[ePayService](#)

D434155

PayPal

Donate



UserG
Proxy & Fi

Сообщение сер

Данный сайт за

<http://vk.com/wi>
app=0&width=2

You Tube - видеOVERсия урока

Дерево-список строить мы умеем, теперь посмотрим, как с ним можно взаимодействовать. Нам предоставлена возможность обрабатывать следующие события: нажатие на группу, нажатие на элемент, сворачивание группы, разворачивание группы.

Создадим проект:

Project name: P0461_ExpandableListEvents

Build Target: Android 2.3.3

Application name: ExpandableListEvents

Package name: ru.startandroid.develop.p0461expandablelistevents

Create Activity: MainActivity

Экран **main.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="fill_parent"
    android:orientation="vertical">
    <LinearLayout
        android:id="@+id/LinearLayout1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:id="@+id/tvInfo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
        </TextView>
        <ExpandableListView
            android:id="@+id/elvMain"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
        </ExpandableListView>
    </LinearLayout>
</LinearLayout>
```

TextView для вывода информации и **ExpandableListView**.

В проекте, рядом с классом **MainActivity** создадим (не Activity) класс **AdapterHelper**. В

него поместим код для заполнения списка, чтобы разгрузить MainActivity.java.

Код **AdapterHelper.java**:

```
package ru.startandroid.develop.p0461expandablelistevents;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import android.content.Context;
import android.widget.SimpleExpandableListAdapter;

public class AdapterHelper {

    final String ATTR_GROUP_NAME= "groupName";
    final String ATTR_PHONE_NAME= "phoneName";

    // названия компаний (групп)
    String[] groups = new String[] {"HTC", "Samsung", "LG"};

    // названия телефонов (элементов)
    String[] phonesHTC = new String[] {"Sensation", "Desire", "Wildfire", "Hero"};
    String[] phonesSams = new String[] {"Galaxy S II", "Galaxy Nexus", "Wave"};
    String[] phonesLG = new String[] {"Optimus", "Optimus Link", "Optimus Black",

    // коллекция для групп
    ArrayList<Map<String, String>> groupData;

    // коллекция для элементов одной группы
    ArrayList<Map<String, String>> childDataItem;

    // общая коллекция для коллекций элементов
    ArrayList<ArrayList<Map<String, String>>> childData;
    // в итоге получится childData = ArrayList<childDataItem>

    // список атрибутов группы или элемента
    Map<String, String> m;

    Context ctx;

    AdapterHelper(Context _ctx) {
        ctx = _ctx;
    }

    SimpleExpandableListAdapter adapter;

    SimpleExpandableListAdapter getAdapter() {

        // заполняем коллекцию групп из массива с названиями групп
        groupData = new ArrayList<Map<String, String>>();
        for (String group : groups) {
            // заполняем список атрибутов для каждой группы
            m = new HashMap<String, String>();
            m.put(ATTR_GROUP_NAME, group); // имя компании
            groupData.add(m);
        }
    }
}
```

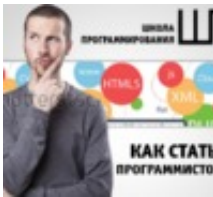
КОМПОНЕНТЫ АВТОМАТИЗАЦИИ

WAGO -
компоненты
автоматизации,
интерфейсные
модули
intradesystems.ru
Адрес и телефон



Нужно сделать Android- приложение

Найди
оптимального
разработчика
под Android
среди сотен
mobile-студий.
inforza.ru



Научиться программировать за 14дн

Интенсивный
курс
программирования
и создания
сайтов.
Сертификат.
Торопись
[Сертификат](#)
[Трудоустройство](#)
[Стажировка](#)
[Отзывы](#)
basics.proglive.ru
Адрес и телефон

```
// список атрибутов групп для чтения
String groupFrom[] = new String[] {ATTR_GROUP_NAME};
// список ID view-элементов, в которые будет помещены атрибуты групп
int groupTo[] = new int[] {android.R.id.text1};

// создаем коллекцию для коллекций элементов
childData = new ArrayList<ArrayList<Map<String, String>>>();

// создаем коллекцию элементов для первой группы
childDataItem = new ArrayList<Map<String, String>>();
// заполняем список атрибутов для каждого элемента
for (String phone : phonesHTC) {
    m = new HashMap<String, String>();
    m.put(ATTR_PHONE_NAME, phone); // название телефона
    childDataItem.add(m);
}
// добавляем в коллекцию коллекций
childData.add(childDataItem);

// создаем коллекцию элементов для второй группы
childDataItem = new ArrayList<Map<String, String>>();
for (String phone : phonesSams) {
    m = new HashMap<String, String>();
    m.put(ATTR_PHONE_NAME, phone);
    childDataItem.add(m);
}
childData.add(childDataItem);

// создаем коллекцию элементов для третьей группы
childDataItem = new ArrayList<Map<String, String>>();
for (String phone : phonesLG) {
    m = new HashMap<String, String>();
    m.put(ATTR_PHONE_NAME, phone);
    childDataItem.add(m);
}
childData.add(childDataItem);

// список атрибутов элементов для чтения
String childFrom[] = new String[] {ATTR_PHONE_NAME};
// список ID view-элементов, в которые будет помещены атрибуты элементов
int childTo[] = new int[] {android.R.id.text1};

adapter = new SimpleExpandableListAdapter(
    ctx,
    groupData,
    android.R.layout.simple_expandable_list_item_1,
    groupFrom,
    groupTo,
    childData,
    android.R.layout.simple_list_item_1,
    childFrom,
    childTo);

return adapter;
}

String getGroupText(int groupPos) {
    return ((Map<String, String>)(adapter.getGroup(groupPos))).get(ATTR_GROUP_NAME);
}
```

```
String getChildText(int groupPos, int childPos) {
    return ((Map<String,String>)(adapter.getChild(groupPos, childPos))).get(ATTR_
}

String getGroupChildText(int groupPos, int childPos) {
    return getGroupText(groupPos) + " " + getChildText(groupPos, childPos);
}
}
```

Код создания адаптера полностью заимствован с прошлого урока. Чтобы получить адаптер нам надо будет просто вызвать метод **getAdapter**.

У класса есть **конструктор**, через который мы передаем объекту ссылку на **context**. Context нам понадобится, чтобы создать адаптер. Адаптеру же в свою очередь context нужен, например, для доступа к LayoutInflater.

В конце класса находятся методы, которые возвращают нам названия групп и элементов из коллекций по номеру группы или номеру элемента. Для этого используем методы адаптера **getGroup** и **getChild**, приводим их к **Map** и извлекаем значение **атрибута** с именем компании или телефона.

Пишем код в **MainActivity.java**:

```
package ru.startandroid.develop.p0461expandablelistevents;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ExpandableListView;
import android.widget.ExpandableListView.OnChildClickListener;
import android.widget.ExpandableListView.OnGroupClickListener;
import android.widget.ExpandableListView.OnGroupCollapseListener;
import android.widget.ExpandableListView.OnGroupExpandListener;
import android.widget.SimpleExpandableListAdapter;
import android.widget.TextView;

public class MainActivity extends Activity {

    final String LOG_TAG = "myLogs";

    ExpandableListView elvMain;
    AdapterHelper ah;
    SimpleExpandableListAdapter adapter;
    TextView tvInfo;

    /** Called when the activity is first created. */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        tvInfo = (TextView) findViewById(R.id.tvInfo);
```

```

// создаем адаптер
ah = new AdapterHelper(this);
adapter = ah.getAdapter();

elvMain = (ExpandableListView) findViewById(R.id.elvMain);
elvMain.setAdapter(adapter);

// нажатие на элемент
elvMain.setOnChildClickListener(new OnChildClickListener() {
public boolean onChildClick(ExpandableListView parent, View v,
    int groupPosition, int childPosition, long id) {
    Log.d(LOG_TAG, "onChildClick groupPosition = " + groupPosition +
        " childPosition = " + childPosition +
        " id = " + id);
    tvInfo.setText(ah.getChildText(groupPosition, childPosition));
    return false;
}
});

// нажатие на группу
elvMain.setOnGroupClickListener(new OnGroupClickListener() {
public boolean onGroupClick(ExpandableListView parent, View v,
    int groupPosition, long id) {
    Log.d(LOG_TAG, "onGroupClick groupPosition = " + groupPosition +
        " id = " + id);
    // блокируем дальнейшую обработку события для группы с позицией 1
    if (groupPosition == 1) return true;

    return false;
}
});

// сворачивание группы
elvMain.setOnGroupCollapseListener(new OnGroupCollapseListener() {
public void onGroupCollapse(int groupPosition) {
    Log.d(LOG_TAG, "onGroupCollapse groupPosition = " + groupPosition);
    tvInfo.setText("Свернули " + ah.getGroupText(groupPosition));
}
});

// разворачивание группы
elvMain.setOnGroupExpandListener(new OnGroupExpandListener() {
public void onGroupExpand(int groupPosition) {
    Log.d(LOG_TAG, "onGroupExpand groupPosition = " + groupPosition);
    tvInfo.setText("Развернули " + ah.getGroupText(groupPosition));
}
});

// разворачиваем группу с позицией 2
elvMain.expandGroup(2);
}

```

Благодаря классу **AdapterHelper**, код создания адаптера занял всего две строчки: **создание** объекта и вызов метода **getAdapter**. Далее присваиваем адаптер списку и добавляем **обработчики**:

1) [OnChildClickListener](#) - нажатие на элемент

Метод

public boolean **onChildClick**(ExpandableListView parent, View v, int groupPosition, int childPosition, long id), где

parent – ExpandableListView с которым работаем

v – View элемента

groupPosition – позиция группы в списке

childPosition – позиция элемента в группе

id – id элемента

Мы выводим в лог позицию и id. А в **TextView** сверху от списка выводим **текст** нажатого **элемента** и его **группы**, который получаем с помощью методов AdapterHelper.

Метод должен вернуть **boolean**. Если мы возвращаем **true** – это значит, мы сообщаем, что сами полностью **обработали событие** и оно **не пойдет в дальнейшие обработчики** (если они есть). Если возвращаем **false** – значит, мы позволяем **событию идти дальше**.

2) [OnGroupClickListener](#) – нажатие на группу

Метод

public boolean **onGroupClick**(ExpandableListView parent, View v, int groupPosition, long id), где

parent – ExpandableListView с которым работаем

v – View элемента

groupPosition – позиция группы в списке

id – id группы

Мы выводим в лог позицию и id группы.

Этот метод также должен вернуть **boolean**. Мы будем возвращать true, если позиция группы = 1, иначе - false. Т.е. для этой группы мы блокируем дальнейшую обработку события. Далее увидим, что нам это даст.

3) [OnGroupCollapseListener](#) – сворачивание группы

Метод

onGroupCollapse(int groupPosition), где groupPosition – позиция группы, которую свернули

4) [OnGroupExpandListener](#) – разворачивание группы

Метод

onGroupExpand(int groupPosition), где groupPosition – позиция группы, которую развернули

И в конце кода MainActivity мы разворачиваем группу с позицией 2, используя метод

[expandGroup](#).

Все сохраним и запускаем.

Как видим, группа LG сразу развернута. Это сработала команда **expandGroup** в конце кода.

Если посмотреть в лог, то видим

```
onGroupExpand groupPosition = 2
```

Т.е. отработало событие разворачивания группы с позицией 2.

Нажмем, например, на Optimus Link. Смотрим лог:

```
onChildClick groupPosition = 2 childPosition = 1 id = 1
```

Не забываем, что позиция считается с нуля. Группа с позицией 2 – LG, элемент с позицией 1 – Optimus Link, все верно.

Смотрим TextView сверху экрана, он считал из адаптера значение атрибута и отобразил его.

Теперь попробуем свернуть группу LG, нажмем на нее. Смотрим лог:

```
onGroupClick groupPosition = 2 id = 2  
onGroupCollapse groupPosition = 2
```

Сначала отработал **onGroupClick** – нажатие на группу, а потом **onGroupCollapse** – сворачивание группы. TextView наверху экрана оповестил о том, что свернули группу LG.

Снова развернем группу LG. Лог:

```
onGroupClick groupPosition = 2 id = 2  
onGroupExpand groupPosition = 2
```

Нажатие на группу и разворачивание. Обратите внимание, что при программном разворачивании, события нажатия не было, только разворот.

Теперь попробуем развернуть группу с позицией 1 – Samsung. Группа не разворачивается. Смотрим лог:

```
onGroupClick groupPosition = 1 id = 1
```

Событие нажатия есть, а вот обработчик разворачивания не вызывается. Это

происходит из-за строки

```
// блокируем дальнейшую обработку события для группы с позицией 1  
if (groupPosition == 1) return true;
```

Мы для группы с позицией 1 блокируем дальнейшую обработку события и оно не уходит в обработчики разворачивания или сворачивания. Поэтому и не срабатывает **onGroupExpand**.

В итоге эти 4 обработчика позволяют вам определять, как пользователь взаимодействует с деревом.

► [Обсудить на форуме \[71 replies\]](#)

[< Назад](#) [Вперёд >](#)

Спасибо за вашу поддержку сайта!

100 руб.



Поддержать

Перевод проекту [StartAndroid](#)

▲ TOP

T3 Framework
FAST. FLEXIBLE. POWERFUL.

При использовании материалов сайта ссылка на startandroid.ru обязательна.

