[События ExpandableListView](https://startandroid.ru/ru/uroki/vse-uroki-spiskom/88-urok-46-sobytija-expandablelistview.html" \o "Урок 46. События ExpandableListView)

- обрабатываем события дерева-списка

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**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | <?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**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125 | 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", "Optimus One"};      // 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 аттрибутов 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 аттрибутов для каждой группы            m = new HashMap<String, String>();              m.put(ATTR\_GROUP\_NAME, group); // имя компании              groupData.add(m);          }            // список аттрибутов групп для чтения          String groupFrom[] = new String[] {ATTR\_GROUP\_NAME};          // список ID view-элементов, в которые будет помещены аттрибуты 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 аттрибутов 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 аттрибутов элементов для чтения          String childFrom[] = new String[] {ATTR\_PHONE\_NAME};          // список ID view-элементов, в которые будет помещены аттрибуты 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\_PHONE\_NAME);    }      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**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83 | 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.getGroupChildText(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](http://developer.android.com/reference/android/widget/ExpandableListView.OnChildClickListener.html" \t "_blank) - 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](http://developer.android.com/reference/android/widget/ExpandableListView.OnGroupClickListener.html" \t "_blank) – 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 будет 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](http://developer.android.com/reference/android/widget/ExpandableListView.OnGroupCollapseListener.html" \t "_blank) – сворачивание группы

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

4) [OnGroupExpandListener](http://developer.android.com/reference/android/widget/ExpandableListView.OnGroupExpandListener.html" \t "_blank) – 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](http://developer.android.com/reference/android/widget/ExpandableListView.html" \l "expandGroup(int)" \t "_blank).

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 строчки

|  |  |
| --- | --- |
| 1  2 | // 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**.