[События в ListView](https://startandroid.ru/ru/uroki/vse-uroki-spiskom/85-urok-44-sobytija-v-listview.html" \o "Урок 44. События в ListView)

- рассматриваем события ListView: нажатие - onItemClick, выделение - onItemSelect, прокрутка - onScroll

При взаимодействии со списком может возникнуть необходимость обрабатывать события – нажатие на пункт и прокрутка. Попробуем это сделать.

**Создадим** **проект:**

**Project name**: P0441\_SimpleListEvents  
**Build Target**: Android 2.3.3  
**Application name**: SimpleListEvents  
**Package name**: ru.startandroid.develop.p0441simplelistevents  
**Create Activity**: MainActivity

Нарисуем 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">      <ListView          android:id="@+id/lvMain"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content">      </ListView>  </LinearLayout> |

На экране только **ListView**.

Так же, как и на прошлом уроке добавим список имен в ресурс **res/values/strings.xml**:

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <resources>      <string name="hello">Hello World, MainActivity!</string>      <string name="app\_name">SimpleListEvents</string>      <string-array name="names">          <item>Иван</item>          <item>Марья</item>          <item>Петр</item>          <item>Антон</item>          <item>Даша</item>          <item>Борис</item>          <item>Костя</item>          <item>Игорь</item>          <item>Анна</item>          <item>Денис</item>          <item>Вадим</item>          <item>Ольга</item>          <item>Сергей</item>      </string-array>  </resources> |

Пишем код **MainActivity.java**:

|  |
| --- |
| package ru.startandroid.develop.p0441simplelistevents;    import android.app.Activity;  import android.os.Bundle;  import android.util.Log;  import android.view.View;  import android.widget.AdapterView;  import android.widget.AdapterView.OnItemClickListener;  import android.widget.AdapterView.OnItemSelectedListener;  import android.widget.ArrayAdapter;  import android.widget.ListView;    public class MainActivity extends Activity {      final String LOG\_TAG = "myLogs";      ListView lvMain;      /\*\* Called when the activity is first created. \*/    public void onCreate(Bundle savedInstanceState) {      super.onCreate(savedInstanceState);      setContentView(R.layout.main);        lvMain = (ListView) findViewById(R.id.lvMain);        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(          this, R.array.names, android.R.layout.simple\_list\_item\_1);      lvMain.setAdapter(adapter);        lvMain.setOnItemClickListener(new OnItemClickListener() {        public void onItemClick(AdapterView<?> parent, View view,            int position, long id) {          Log.d(LOG\_TAG, "itemClick: position = " + position + ", id = "              + id);        }      });        lvMain.setOnItemSelectedListener(new OnItemSelectedListener() {        public void onItemSelected(AdapterView<?> parent, View view,            int position, long id) {          Log.d(LOG\_TAG, "itemSelect: position = " + position + ", id = "              + id);        }          public void onNothingSelected(AdapterView<?> parent) {          Log.d(LOG\_TAG, "itemSelect: nothing");        }      });      }  } |

Смотрим код. Мы находим экранные элементы, создаем и присваиваем списку адаптер. Далее списку мы присваиваем два обработчика событий:

1) **OnItemClickListener**– обрабатывает **нажатие**на пункт списка

Предоставляет нам метод [onItemClick(AdapterView<?> parent, View view, int position, long id)](http://developer.android.com/reference/android/widget/AdapterView.OnItemClickListener.html" \l "onItemClick(android.widget.AdapterView%3C?%3E,%20android.view.View,%20int,%20long)" \t "_blank), где

**parent**– View-родитель для нажатого пункта, в нашем случае - ListView   
**view**– это нажатый пункт, в нашем случае – TextView из android.R.layout.simple\_list\_item\_1  
**position**– порядковый номер пункта в списке  
**id**– идентификатор элемента,

Мы в лог будем выводить **id**и **position**для элемента, на который нажали.

2) **OnItemSelectedListener**– обрабатывает **выделение** пунктов списка (не check, как на прошлом уроке)

Предоставляет нам метод [onItemSelected](http://developer.android.com/reference/android/widget/AdapterView.OnItemSelectedListener.html" \l "onItemSelected(android.widget.AdapterView%3C?%3E,%20android.view.View,%20int,%20long)" \t "_blank) полностью аналогичен по параметрам методу **onItemClick**описанному выше. Не буду повторяться.

Также есть метод [onNothingSelected](http://developer.android.com/reference/android/widget/AdapterView.OnItemSelectedListener.html" \l "onNothingSelected(android.widget.AdapterView%3C?%3E)" \t "_blank) – когда список **теряет выделение** пункта и ни один пункт не выделен.

Все сохраним и запустим приложение.

Ткнем какой-нибудь элемент, например - Петр. Смотрим лог:

*itemClick: position = 2, id = 2*

Все верно. Т.к. позиция считается не с единицы, а с **нуля**– Петр имеет позицию 2. (В нашем случае id равен position. Я пока не встречал случаев id != position, но наверняка они есть)

Теперь покрутите колесо мышки или понажимайте клавиши вверх вниз на клавиатуре. Видно что идет **визуальное выделение**элементов списка.

А в логах мы видим такие записи:

*itemSelect: position = 2, id = 2  
itemSelect: position = 3, id = 3  
itemSelect: position = 4, id = 4  
itemSelect: position = 5, id = 5  
itemSelect: position = 4, id = 4  
itemSelect: position = 3, id = 3  
itemSelect: position = 2, id = 2*

Т.е. обработчик фиксирует какой пункт **выделен**. Честно говоря, я не очень понимаю как можно использовать такое выделение. Но обработчик для него есть и я решил про него рассказать. Пусть будет.

Снова нажмем теперь на любой пункт списка, мы видим, что выделение пропало. Логи:

*itemSelect: nothing  
itemClick: position = 3, id = 3*

Ничего не выделено и нажат пункт с позицией 3.

Давайте добавим к списку еще один обработчик:

|  |
| --- |
| lvMain.setOnScrollListener(new OnScrollListener() {    public void onScrollStateChanged(AbsListView view, int scrollState) {      // Log.d(LOG\_TAG, "scrollState = " + scrollState);    }      public void onScroll(AbsListView view, int firstVisibleItem,        int visibleItemCount, int totalItemCount) {      Log.d(LOG\_TAG, "scroll: firstVisibleItem = " + firstVisibleItem          + ", visibleItemCount" + visibleItemCount          + ", totalItemCount" + totalItemCount);    }  }); |

**OnScrollListener** – обрабатывает прокрутку списка.

Методы:

1) [onScrollStateChanged(AbsListView view, int scrollState)](http://developer.android.com/reference/android/widget/AbsListView.OnScrollListener.html" \l "onScrollStateChanged(android.widget.AbsListView,%20int)" \t "_blank) - обработка состояний прокрутки

**view**– это прокручиваемый элемент, т.е. ListView  
**scrollState**– состояние списка. Может принимать три значения:

**SCROLL\_STATE\_IDLE**= 0, список закончил прокрутку  
**SCROLL\_STATE\_TOUCH\_SCROLL**= 1, список начал прокрутку  
**SCROLL\_STATE\_FLING**= 2, список «катнули», т.е. при прокрутке отпустили палец и прокрутка дальше идет «по инерции»

Вывод в лог я пока закаментил, чтобы не мешалось. Чуть позже раскаментим.

2) [onScroll(AbsListView view, int firstVisibleItem, int visibleItemCount, int totalItemCount)](http://developer.android.com/reference/android/widget/AbsListView.OnScrollListener.html" \l "onScroll(android.widget.AbsListView,%20int,%20int,%20int)" \t "_blank) - обработка прокрутки

**view**– прокручиваемый элемент  
**firstVisibleItem**– первый видимый на экране пункт списка  
**visibleItemCount**– сколько пунктов видно на экране  
**totalItemCount**– сколько всего пунктов в списке

Причем для параметров **firstVisibleItem**и **visibleItemCount**пункт считается видимым на экране даже если он виден не полностью.

Все сохраним и запустим.

Теперь потаскайте список туда-сюда курсором (как будто пальцем) и смотрите логи. Там слишком много всего выводится. Я не буду здесь выкладывать. Но принцип понятен – меняется первый видимый пункт (firstVisibleItem) и может на единицу меняться кол-во видимых пунктов (visibleItemCount).

Теперь закоментируем вывод в лог в методе **onScroll**(чтобы не спамил нам лог) и раскаментим в **onScrollStateChanged**.

|  |
| --- |
| lvMain.setOnScrollListener(new OnScrollListener() {    public void onScrollStateChanged(AbsListView view, int scrollState) {      Log.d(LOG\_TAG, "scrollState = " + scrollState);    }      public void onScroll(AbsListView view, int firstVisibleItem,        int visibleItemCount, int totalItemCount) {      //Log.d(LOG\_TAG, "scroll: firstVisibleItem = " + firstVisibleItem      //    + ", visibleItemCount" + visibleItemCount      //    + ", totalItemCount" + totalItemCount);    }  }); |

Сохраняем, запускаем.

Схватим список, немного потягаем туда сюда и отпустим. Смотрим лог:

*scrollState = 1  
scrollState = 0*

Отработали два события – список **начал**прокрутку, список **закончил**прокрутку.

Попробуем взять список, «катнуть» его и отпустить.

*scrollState = 1  
scrollState = 2  
scrollState = 0*

Видим три события – прокрутка **началась**, список «**катнули**», прокрутка **закончилась**.

Полный код урока:

|  |
| --- |
| package ru.startandroid.develop.p0441simplelistevents;    import android.app.Activity;  import android.os.Bundle;  import android.util.Log;  import android.view.View;  import android.widget.AbsListView;  import android.widget.AbsListView.OnScrollListener;  import android.widget.AdapterView;  import android.widget.AdapterView.OnItemClickListener;  import android.widget.AdapterView.OnItemSelectedListener;  import android.widget.ArrayAdapter;  import android.widget.ListView;    public class MainActivity extends Activity {      final String LOG\_TAG = "myLogs";      ListView lvMain;      /\*\* Called when the activity is first created. \*/    public void onCreate(Bundle savedInstanceState) {      super.onCreate(savedInstanceState);      setContentView(R.layout.main);        lvMain = (ListView) findViewById(R.id.lvMain);        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(          this, R.array.names, android.R.layout.simple\_list\_item\_1);      lvMain.setAdapter(adapter);        lvMain.setOnItemClickListener(new OnItemClickListener() {        public void onItemClick(AdapterView<?> parent, View view,            int position, long id) {          Log.d(LOG\_TAG, "itemClick: position = " + position + ", id = "              + id);        }      });        lvMain.setOnItemSelectedListener(new OnItemSelectedListener() {        public void onItemSelected(AdapterView<?> parent, View view,            int position, long id) {          Log.d(LOG\_TAG, "itemSelect: position = " + position + ", id = "              + id);        }          public void onNothingSelected(AdapterView<?> parent) {          Log.d(LOG\_TAG, "itemSelect: nothing");        }      });        lvMain.setOnScrollListener(new OnScrollListener() {        public void onScrollStateChanged(AbsListView view, int scrollState) {          Log.d(LOG\_TAG, "scrollState = " + scrollState);        }          public void onScroll(AbsListView view, int firstVisibleItem,            int visibleItemCount, int totalItemCount) {          Log.d(LOG\_TAG, "scroll: firstVisibleItem = " + firstVisibleItem              + ", visibleItemCount" + visibleItemCount              + ", totalItemCount" + totalItemCount);        }      });      }  } |