

# Farhan's Blogs & Portal

*My Blogs, Article etc.*



[Home](#) [Tutorials](#) [Other Links](#) [Articles/News](#) [Visitors](#) [About](#)

[← Custom Dialog Tutorial for Android](#)

[Manipulating Incoming & Outgoing Calls Tutorial For Android →](#)

## List View Tutorial For Android

Posted on [October 14, 2010](#)

List View Tutorial is used to show data in tabular form as shown in picture.

Here is the complete [source code](#).



Search

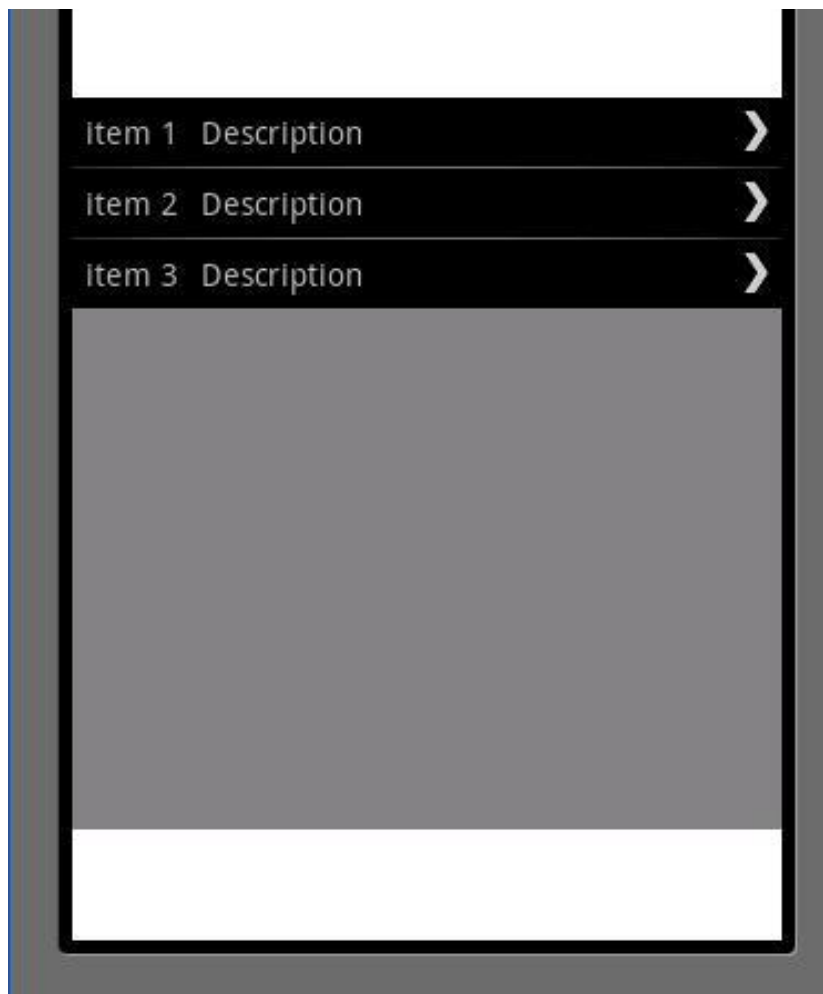
### Email Subscription

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Join 36 other followers

Subscribe

Visitors Count



### list\_item.xml

In list\_item.xml i simply designed a cell of list which contains two text views(name & description) and imageview(forward arrow).

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:padding="6px"
    android:layout_height="67px" android:id="@+id/rlt_main"
    android:background="#000000">
    <TextView android:layout_width="wrap_content"
```

### Recent Posts

- [TabLayout/TabHost Tutorial For Android \(Reusing Layout\)](#)
- [Working with SQLite DB Tutorial](#)
- [DataBase-CRUD Tutorial for Android](#)
- [Manipulating Incoming & Outgoing Calls Tutorial For Android](#)
- [List View Tutorial For Android](#)

### Archives

- [November 2010](#)
- [October 2010](#)
- [September 2010](#)
- [August 2010](#)

### Categories

- [Android](#)
- [Android Links](#)
- [Articles/News](#)
- [Other Links](#)
- [Sci-Tech](#)
- [Tutorials](#)
- [Uncategorized](#)

### Tags Wall

[AlertDialog](#) **Android**  
[Android Dialogs](#)  
**Android Tutorials**

[Array Adapter Audio](#)

[AutoComplete Cricket CRUD](#)

[Custom Dialogs DB Linear Layout](#)

```

        android:layout_height="wrap_content" android:id="@+id/tv_name"
        android:text="Name" android:layout_centerVertical="true"></TextView>
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Description"
        android:id="@+id/tv_description" android:layout_centerVertical="true"
        android:layout_toRightOf="@+id/tv_name"
        android:layout_marginLeft="10px"></TextView>
    <ImageView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_centerVertical="true"
        android:id="@+id/iv_forward" android:background="@drawable/forward_arrow"
        android:layout_alignParentRight="true"></ImageView>
</RelativeLayout>

```

## main.xml

In main.xml i designed the layout contain header and footer(relative layout) having height 50 px. In between them there is relative layout containing list view used to show items list.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout android:id="@+id/RelativeLayout01"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <RelativeLayout android:id="@+id/RelativeLayout02"
        android:layout_height="50px" android:layout_width="fill_parent"
        android:background="#ffffff"></RelativeLayout>
    <RelativeLayout android:id="@+id/RelativeLayout03"
        android:layout_width="fill_parent" android:layout_height="fill_parent"
        android:background="#808080" android:layout_above="@+id/RelativeLayout04"
        android:layout_below="@+id/RelativeLayout02">
        <ListView android:layout_width="wrap_content"
            android:layout_height="wrap_content" android:id="@+id/list_view"></ListView>
    </RelativeLayout>
    <RelativeLayout android:id="@+id/RelativeLayout04"
        android:layout_height="50px" android:layout_width="fill_parent"
        android:layout_alignParentBottom="true"
        android:background="#ffffff"></RelativeLayout>
</RelativeLayout>

```

## Splash.java

In Splash activity i bind the itemslist coming from itemBO.getItems() method.

[Class ListAdapter](#)
[List View](#)
[Maps](#)  
[API Key](#)
[Master Lay out](#)  
[onCallStateChanged](#)  
[PhoneStateListener](#)
[PopUps](#)  
[Progress Bar](#)
[Reusing Layout](#)  
[SQLite](#)
[SQLite Manager](#)
[Swing](#)  
[TabHost](#)
[TabLayout](#)  
[Telephony Manager](#)
[Thread](#)
[Toast](#)  
[Video](#)

## Meta

- [Register](#)
- [Log in](#)
- [Entries RSS](#)
- [Comments RSS](#)
- [WordPress.com](#)

## Top Posts

- [Progress Bar Tutorial For Android](#)
- [Manipulating Incoming & Outgoing Calls Tutorial For Android](#)
- [TabLayout/TabHost Tutorial For Android \(Reusing Layout\)](#)
- [List View Tutorial For Android](#)

## Top Clicks

- [megaupload.com/?d=GIYGDKZ...](#)
- [megaupload.com/?d=PA4H8ZC...](#)
- [megaupload.com/?d=8CNM55G...](#)
- [krvarma.com/posts/android...](#)
- [megaupload.com/?d=JBHQMEU...](#)

In order to bind list i have to make ArrayAdapter Class ArrayList<ItemBO>, Context attributes and Constructor and getView() methods.

In getView() method i inflate the list\_item layout and implement onClickListner to ListCell to apply action when user clicks on list cell which starts new Activity and pass ItemName to it in Intent.

```
public class Splash extends Activity implements OnClickListener {

    private ListView listview;
    private ArrayList mListItem;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        listview = (ListView) findViewById(R.id.list_view);
        mListItem = ItemBO.getItems();
        listview.setAdapter(new ListAdapter(Splash.this, R.id.list_view,
            mListItem));
    }

    @Override
    public void onClick(View v) {
    }

    // ***ListAdapter***
    private class ListAdapter extends ArrayAdapter { //--CloneChangeRequired
        private ArrayList mList; //--CloneChangeRequired
        private Context mContext;

        public ListAdapter(Context context, int textViewResourceId,
            ArrayList list) { //--CloneChangeRequired
            super(context, textViewResourceId, list);
            this.mList = list;
            this.mContext = context;
        }

        public View getView(int position, View convertView, ViewGroup parent) {
            View view = convertView;
            try {
                if (view == null) {
```

```

        LayoutInflater vi = (LayoutInflater)
getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        view = vi.inflate(R.layout.list_item, null); //--
CloneChangeRequired(list_item)
    }
    final ItemBO listItem = mList.get(position); //--CloneChangeRequired
    if (listItem != null) {
        // setting list_item views
        ((TextView) view.findViewById(R.id.tv_name))
            .setText(listItem.getName());
        view.setOnClickListener(new OnClickListener() {
            public void onClick(View arg0) { //--clickOnListItem
                Intent myIntent = new Intent(Splash.this,
                    Activity2.class);
                myIntent.putExtra("NAME", listItem.getName());
                startActivity(myIntent);
                finish();
            }
        });
    }
} catch (Exception e) {
    Log.i(Splash.ListAdapter.class.toString(), e.getMessage());
}
return view;
}
}
}

```

## Activity2.java

In Activity2 i just show that which item of items list is clicked and allow to go back.

```

public class Activity2 extends Activity {
    private Context context;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        RelativeLayout relativeLayout = new RelativeLayout(this);
        Button button = new Button(this);
        Bundle bundle = getIntent().getExtras();
        if (bundle != null) {
            button.setText("Item name = " + bundle.getString("NAME")
                + " --- Go Back ");
        }
    }
}

```

```

    } else {
        button.setText("Go Back");
    }
    context = this;
    button.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            startActivity(new Intent(context, Splash.class));
            finish();
        }
    });
    relativeLayout.addView(button);
    LayoutParams params = new LayoutParams(LayoutParams.FILL_PARENT,
        LayoutParams.FILL_PARENT);
    setContentView(relativeLayout, params);
}
}

```

## ItemBO.java

It is the itemBO class containing name, description attributes and getItems() method.

```

public class ItemBO {
    private String name;
    private String description;
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getDescription() { return description; }
    public void setDescription(String description) { this.description = description; }
    // / -----
    public static ArrayList.getItems() {
        ArrayList list = new ArrayList();
        ItemBO item;

        item = new ItemBO();
        item.setName("item 1");
        item.setDescription("desc 1");
        list.add(item);

        item = new ItemBO();
        item.setName("item 2");
        item.setDescription("desc 2");
        list.add(item);
    }
}

```

```
item = new ItemBO();
item.setName("item 3");
item.setDescription("desc 3");
list.add(item);

return list;
}
```

Here is the complete [source code](#).

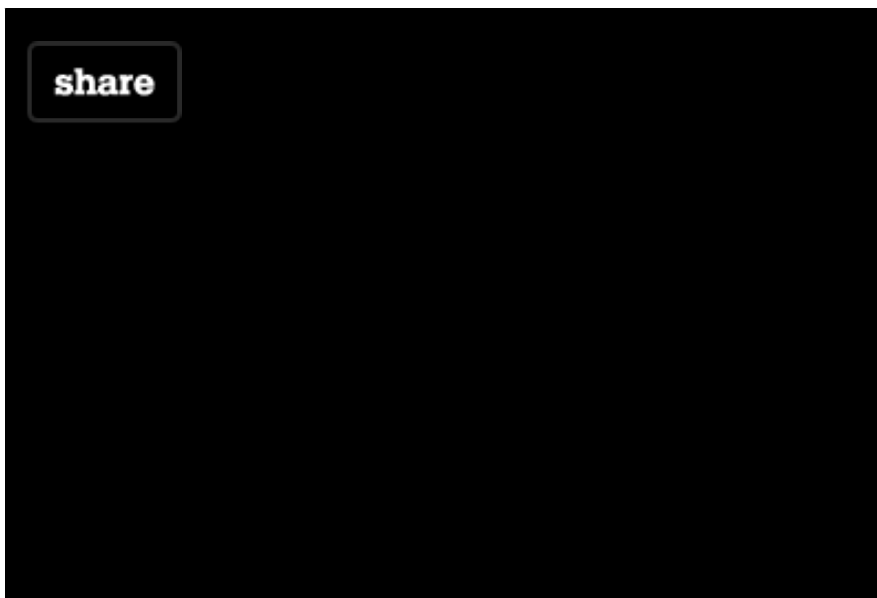
You can also extend ListView Class and make your own Generic List. [Here its source code](#).

**Other Links**

<http://ykyuen.wordpress.com/2010/01/03/android-simple-listview-using-simpleadapter/>



[About these ads](#)



Rate this:



15 Votes

Like this:



One blogger likes this.



This entry was posted in [Android](#), [Tutorials](#) and tagged [Android](#), [Android Tutorials](#), [List View](#), [ListAdapter](#). Bookmark the [permalink](#).

[← Custom Dialog Tutorial for Android](#)

[Manipulating Incoming & Outgoing Calls Tutorial For Android →](#)

## 5 Responses to *List View Tutorial For Android*



**david** says:

November 30, 2010 at 10:27 am

am i supposed to do any thing with the string.xml file and drawables folder?pls help!



[Reply](#)

---



**MB** *says:*

January 6, 2011 at 9:45 pm

Aren't you missing a cast in this line?

```
final ItemBO listItem = mList.get(position);
```

[Reply](#)

---



**sunil** *says:*

February 23, 2011 at 9:32 am

Thanks.

[Reply](#)

---



**[Janar Jürisson](#)** *says:*

March 14, 2011 at 9:50 am

Great.

Helped me out here with the custom lists.

Simple and worked almost instantly for me.

[Reply](#)

---



**charly** *says:*



February 10, 2012 at 11:30 am

Thank you! really useful !!

But I want to start a dialog instead of Activity2. The dialog needs a context, I cannot use a non-static variable in the onClick listener, and the context cannot be static. Then, how I can start a dialog?

[Reply](#)

## Leave a Reply

Enter your comment here...

---

**Farhan's Blogs & Portal**

Theme: Twenty Ten  Blog at WordPress.com.



 Follow

Follow “Farhan's Blogs & Portal”

Get every new post delivered to your Inbox.

Join 36 other followers

Enter your email address