

Program the Mind!

Home Tutorials Examples Contact Us

Search...



Examples Android ListView

ANDROID: CONTEXTUAL ACTION BAR FOR LISTVIEW ITEM DELETION USING ACTIONBARSHERLOCK

27 July 2013 By Nithya Vasudevan 8,128 views 3 Comments

19 SHARES

Create Website on Google

cloud.google.com/appengine/

Build and run your website using Google App Engine



Prior to Android 3.0, the long press gesture (a touch that's held in the same position for a moment) was used to display contextual actions for a given data item in a contextual menu. This pattern changed with Android 3.0. The long press gesture is now used to select data, combining contextual actions and selection management functions for selected data into a new element called the contextual action bar (CAB).

Reference: <http://developer.android.com/design/patterns/selection.html>

1. What is CAB?

The selection CAB is a temporary action bar that overlays your app's current action bar while data is selected. It appears after the user long presses on a selectable data item. Select additional data items by touching them. It is recommended to use when you perform actions with selected data like plain text or data items from ListView or GridView components. **The action mode is disabled and the contextual action bar disappears when the user deselects all items, presses the BACK button, or selects the checkmark button on the left side of the bar.**

BACK TO TOP

2. Implementing CAB for ListView item selection

- If you're using Android 3.0 or higher there is a [detailed description of CAB setup on each selection method](#) - whether contextual action should be performed on a single selected item or on a group of selected items.
- Although it is available from Android 3.0 it is a good practice to use it in apps working on earlier API versions. If your app works on android version 2.x and higher the best approach is to use [ActionBarSherlock](#) library. It has its own contextual action bar implementation, which is easy to set up. But it doesn't support the native ListView integration, so you need to control CAB's lifecycle by yourself.

You can invoke contextual action mode based on one of the two events or both.

1. The user selects a checkbox or similar UI component within the view.
2. The user performs a long-click on the view.

This tutorial explains the second method.

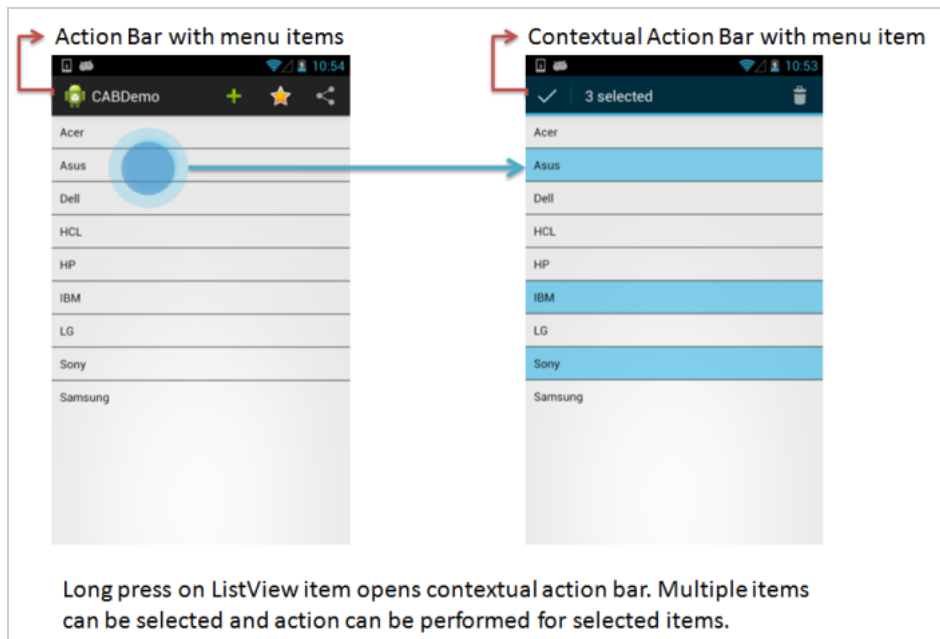
Table of Contents

1 What is CAB?
2 Implementing CAB for ListView item selection
3 Output
4 Android Project
4.1 strings.xml
4.2 strings_menu.xml
4.3 styles.xml
4.4 Menu XML files
4.4.1 activity_main.xml
4.4.2 context_menu.xml
4.5 XML layout files
4.5.1 activity_main.xml
4.5.2 list_item.xml
4.6 Create Bean class
4.6.1 Laptop.java
4.7 Create Android Custom Adapter class for ListView
4.8 Activity
5 Output
6 Folder Structure

[Like](#)

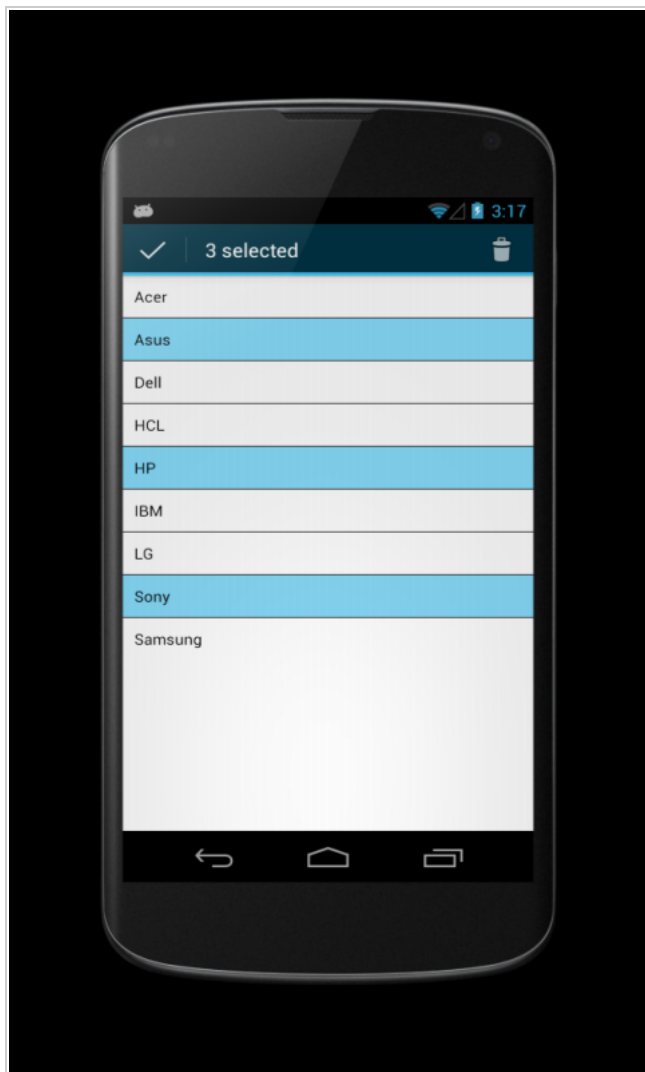
2,176 people like The Open Tutorials.

Facebook social plugin

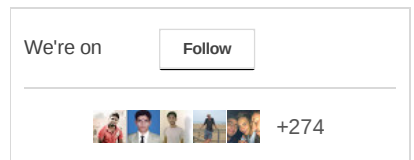


In this tutorial we'll see how to use **Contextual Action Bar for ListView using ActionBarSherlock library** where list items can be selected on **long press gesture**. Here, contextual action bar has **delete menu item**, which **deletes the selected items from ListView**.

3. Output



4. Android Project



Popular Posts

[Pagination in Servlet and JSP](#)

[How to create a simple EJB3 project in Eclipse \(JBoss 7.1\)](#)

[How to create a Servlet with Eclipse and Tomcat](#)

[Android: Custom ListView with Image and Text using ArrayAdapter](#)

[How to create and consume a simple Web Service using JAX WS](#)

[Generate Java class from XML Schema using JAXB 'xjc' command](#)

[How to create a simple Restful Web Service using Jersey JAX RS API](#)

[Android: Expandable List View Example](#)

[How to create EJB3 JPA Project in Eclipse \(JBoss AS 7.1\)](#)

[How to configure Apache Tomcat in Eclipse IDE?](#)

Recent Comments

Anand on [Android: How to send HTTP GET request to Servlet using Apache Http Client](#)

yosef birhane on [Android: Expandable List View Example](#)

Akash on [Getting checkbox values from HTML form in Servlet](#)

Dibyendu on [How to create a simple EJB3 project in Eclipse \(JBoss 7.1\)](#)

Madhav on [How to get list of Installed Apps in Android](#)

Madhav on [How to get list of Installed Apps in Android](#)

Madhav on [How to get list of Installed Apps in Android](#)

Fábio A. Gomes on [Enum Advanced Features JDBC and Apache Derby | Skills421](#)

on [Eclipse DTP: Configure H2 Datasource using Data Source Explorer](#)

gaiapuffo22 on [How to create EJB3 JPA Project in Eclipse \(JBoss AS 7.1\)](#)

Create an Android project and name it as CABDemo. Include ActionBarSherlock as a library project.

4.1. strings.xml

Open res/values/strings.xml and replace it with following content.

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <resources>
03     <string name="app_name">CABDemo</string>
04     <string name="hello_world">Hello world!</string>
05
06     <string-array name="laptops">
07         <item>Acer</item>
08         <item>Asus</item>
09         <item>Dell</item>
10         <item>HCL</item>
11         <item>HP</item>
12         <item>IBM</item>
13         <item>LG</item>
14         <item>Sony</item>
15         <item>Samsung</item>
16     </string-array>
17 </resources>
```

4.2. strings_menu.xml

Create a new strings_menu.xml in res/values/ folder and copy the following content.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <string name="menu_settings">Settings</string>
4     <string name="menu_add">Add</string>
5     <string name="menu_rate">Rate</string>
6     <string name="menu_share">Share</string>
7
8     <string name="menu_delete">Delete</string>
9 </resources>
```

4.3. styles.xml

Open res/values/styles.xml and replace it with following content. Here, we use Action bar Sherlock theme.

```
1 <resources>
2     <style name="AppBaseTheme" parent="Theme.Sherlock">
3     </style>
4
5     <style name="AppTheme" parent="AppBaseTheme">
6     </style>
7 </resources>
```

4.4. Menu XML files

4.4.1. activity_main.xml

Create a new activity_main.xml in res/menu folder and copy the following content. This file defines action bar menu items.

```
01 <menu xmlns:android="http://schemas.android.com/apk/res/android" >
02
03     <item
04         android:id="@+id/menu_add"
05         android:orderInCategory="100"
06         android:showAsAction="ifRoom|withText"
07         android:title="@string/menu_add"
08         android:icon="@android:drawable/ic_input_add"/>
09
10     <item
11         android:id="@+id/menu_rate"
12         android:orderInCategory="200"
13         android:showAsAction="ifRoom|withText"
14         android:title="@string/menu_rate"
15         android:icon="@android:drawable/btn_star_big_on"/>
16
17     <item
18         android:id="@+id/menu_share"
19         android:orderInCategory="300"
20         android:showAsAction="ifRoom|withText"
21         android:title="@string/menu_share"
22         android:icon="@android:drawable/ic_menu_share"/>
23 </menu>
```

4.4.2. context_menu.xml

Create a new **context_main.xml** in **res/menu** folder and copy the following content. This file defines menu items for contextual action bar. We define one menu item which is used to delete items from ListView.

```

1  <menu xmlns:android="http://schemas.android.com/apk/res/android" >
2
3      <item
4          android:id="@+id/menu_delete"
5          android:orderInCategory="100"
6          android:showAsAction="ifRoom|withText"
7          android:title="@string/menu_delete"
8          android:icon="@android:drawable/ic_menu_delete"/>
9  </menu>

```

4.5. XML layout files

This Android contextual action bar example uses two layout files; one for displaying the main layout containing the ListView and other for defining the row layout for ListView item.

4.5.1. activity_main.xml

This application uses XML layout file (activity_main.xml) to display ListView.

Open **activity_main.xml** file in **res/layout** and copy the following content.

```

01  <?xml version="1.0" encoding="utf-8"?>
02  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/anc
03      android:layout_width="match_parent"
04      android:layout_height="fill_parent"
05      android:orientation="vertical" >
06
07      <ListView
08          android:id="@+id/laptopList"
09          android:layout_width="match_parent"
10          android:layout_height="wrap_content"
11          android:divider="#323232"
12          android:dividerHeight="1dp"
13          android:scrollbars="none" >
14  </ListView>
15
16  </LinearLayout>

```

4.5.2. list_item.xml

Create a new layout file **list_item.xml** in **res/layout** and copy the following content. For simplicity, this file contains only a TextView.

```

01  <?xml version="1.0" encoding="utf-8"?>
02  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/anc
03      android:layout_width="match_parent"
04      android:layout_height="match_parent"
05      android:orientation="vertical"
06      android:padding="10dp">
07
08      <TextView
09          android:id="@+id/laptop_name"
10          android:layout_width="wrap_content"
11          android:layout_height="wrap_content"
12          android:textColor="#000000"/>
13
14  </LinearLayout>

```

4.6. Create Bean class

4.6.1. Laptop.java

Create a new Java class "**Laptop.java**" in package "**com.theopentutorials.cabdemo.beans**" and copy the following code.

```

01  package com.theopentutorials.cabdemo.beans;
02
03  public class Laptop {
04      private String brand;
05
06      public Laptop() {
07          super();
08      }
09      public Laptop(String brand) {
10          super();
11          this.brand = brand;

```

```

12     }
13     public String getBrand() {
14         return brand;
15     }
16     public void setBrand(String brand) {
17         this.brand = brand;
18     }
19     @Override
20     public String toString() {
21         return brand;
22     }
23 }

```

4.7. Create Android Custom Adapter class for ListView

Create a new Java class "LaptopListAdapter.java" in package "com.theopentutorials.cabdemo.adapters" and copy the following code.

```

001 package com.theopentutorials.cabdemo.adapters;
002
003 import java.util.List;
004 import android.app.Activity;
005 import android.graphics.Color;
006 import android.util.SparseBooleanArray;
007 import android.view.LayoutInflater;
008 import android.view.View;
009 import android.view.ViewGroup;
010 import android.widget.AdapterView;
011 import android.widget.TextView;
012 import android.widget.Toast;
013
014 import com.theopentutorials.cabdemo.R;
015 import com.theopentutorials.cabdemo.beans.Laptop;
016
017 public class LaptopListAdapter extends ArrayAdapter<Laptop> {
018
019     Activity context;
020     List<Laptop> laptops;
021     private SparseBooleanArray mSelectedItemsIds;
022
023     public LaptopListAdapter(Activity context, int resId, List<Lap
024         super(context, resId, laptops);
025         mSelectedItemsIds = new SparseBooleanArray();
026         this.context = context;
027         this.laptops = laptops;
028     }
029
030     private class ViewHolder {
031         TextView laptopTxt;
032     }
033
034     public View getView(int position, View convertView, ViewGroup
035         ViewHolder holder;
036         if (convertView == null) {
037             LayoutInflater inflater = (LayoutInflater) context
038                 .getSystemService(Activity.LAYOUT_INFLATER_SEF
039             convertView = inflater.inflate(R.layout.list_item, nul
040             holder = new ViewHolder();
041             holder.laptopTxt = (TextView) convertView
042                 .findViewById(R.id.laptop_name);
043             convertView.setTag(holder);
044         } else {
045             holder = (ViewHolder) convertView.getTag();
046         }
047
048         Laptop laptop = getItem(position);
049         holder.laptopTxt.setText(laptop.toString());
050         convertView
051             .setBackgroundColor(mSelectedItemsIds.get(position
052                 : Color.TRANSPARENT);
053
054         return convertView;
055     }
056
057     @Override
058     public void add(Laptop laptop) {
059         laptops.add(laptop);
060         notifyDataSetChanged();
061         Toast.makeText(context, laptops.toString(), Toast.LENGTH_L
062     }
063
064     @Override
065     public void remove(Laptop object) {
066         // super.remove(object);
067         laptops.remove(object);
068         notifyDataSetChanged();
069     }

```

```

070
071     public List<Laptop> getLaptops() {
072         return laptops;
073     }
074
075     public void toggleSelection(int position) {
076         selectView(position, !mSelectedItemsIds.get(position));
077     }
078
079     public void removeSelection() {
080         mSelectedItemsIds = new SparseBooleanArray();
081         notifyDataSetChanged();
082     }
083
084     public void selectView(int position, boolean value) {
085         if (value)
086             mSelectedItemsIds.put(position, value);
087         else
088             mSelectedItemsIds.delete(position);
089
090         notifyDataSetChanged();
091     }
092
093     public int getSelectedCount() {
094         return mSelectedItemsIds.size();
095     }
096
097     public SparseBooleanArray getSelectedIds() {
098         return mSelectedItemsIds;
099     }
100 }

```

4.8. Activity

Open “MainActivity” from the package “com.theopentutorials.cabdemo” and copy the following code.

```

001 package com.theopentutorials.cabdemo;
002
003 import java.util.ArrayList;
004 import java.util.List;
005 import android.app.Activity;
006 import android.os.Bundle;
007 import android.util.SparseBooleanArray;
008 import android.view.Gravity;
009 import android.view.View;
010 import android.widget.AdapterView;
011 import android.widget.AdapterView.OnItemClickListener;
012 import android.widget.AdapterView.OnItemLongClickListener;
013 import android.widget.ListView;
014 import android.widget.Toast;
015
016 import com.actionbarsherlock.app.SherlockFragmentActivity;
017 import com.actionbarsherlock.view.ActionMode;
018 import com.actionbarsherlock.view.Menu;
019 import com.actionbarsherlock.view.MenuInflater;
020 import com.actionbarsherlock.view.MenuItem;
021 import com.theopentutorials.cabdemo.adapters.LaptopListAdapter;
022 import com.theopentutorials.cabdemo.beans.Laptop;
023
024 public class MainActivity extends SherlockFragmentActivity implements
025     OnItemClickListener {
026
027     ListView laptopListView;
028     LaptopListAdapter laptopListAdapter;
029     List<Laptop> laptops = new ArrayList<Laptop>();
030
031     Activity activity = this;
032     private ActionMode mActionMode;
033
034     @Override
035     protected void onCreate(Bundle savedInstanceState) {
036         super.onCreate(savedInstanceState);
037         setContentView(R.layout.activity_main);
038
039         laptopListView = (ListView) findViewById(R.id.laptopList);
040
041         setLaptopList();
042         laptopListAdapter = new LaptopListAdapter(this, R.layout.l
043             laptops);
044         laptopListView.setAdapter(laptopListAdapter);
045         laptopListView.setOnItemClickListener(this);
046         laptopListView.setOnItemLongClickListener(new OnItemLongCl
047
048         public boolean onItemLongClick(AdapterView<?>

```

```

049         View view, int position, long id) {
050             onListItemSelect(position);
051             return true;
052         }
053     });
054 }
055
056 @Override
057 public void onItemClick(AdapterView<?> adapterView, View view,
058     int position, long id) {
059     if (mActionMode == null) {
060         /*no items selected, so perform item click actions
061         * like moving to next activity */
062         Toast toast = Toast.makeText(getApplicationContext(),
063             + (position + 1) + ": " + laptops.get(position),
064             Toast.LENGTH_SHORT);
065         toast.setGravity(Gravity.BOTTOM | Gravity.CENTER_HORIZONTAL);
066         toast.show();
067     } else
068         // add or remove selection for current list item
069         onListItemSelect(position);
070 }
071
072 private void onListItemSelect(int position) {
073     laptopListAdapter.toggleSelection(position);
074     boolean hasCheckedItems = laptopListAdapter.getSelectedCount() > 0;
075
076     if (hasCheckedItems && mActionMode == null)
077         // there are some selected items, start the actionMode
078         mActionMode = startActionMode(new ActionModeCallback() {
079             @Override
080             public void onActionItemClicked(ActionMode mode, MenuItem item) {
081                 // there no selected items, finish the actionMode
082                 mActionMode.finish();
083             }
084         });
085     if (mActionMode != null)
086         mActionMode.setTitle(String.valueOf(laptopListAdapter
087             .getSelectedCount()) + " selected");
088 }
089
090 private void setLaptopList() {
091     String[] blogs = getResources().getStringArray(R.array.laptop_brands);
092     for (String brand : blogs) {
093         Laptop laptop = new Laptop(brand);
094         laptops.add(laptop);
095     }
096 }
097
098 @Override
099 public boolean onCreateOptionsMenu(Menu menu) {
100     // Inflate the menu; this adds items to the action bar if
101     MenuInflater inflater = getSupportMenuInflater();
102     inflater.inflate(R.menu.activity_main, menu);
103     return true;
104 }
105
106 @Override
107 public boolean onOptionsItemSelected(MenuItem item) {
108     if (item.getItemId() == R.id.menu_add) {
109         // adds item to listview
110     } else if (item.getItemId() == R.id.menu_share) {
111         // share your application by using share intent
112     } else if (item.getItemId() == R.id.menu_rate) {
113         // add rate feature to your application by launching market
114     }
115     return true;
116 }
117
118 private class ActionModeCallback implements ActionMode.Callback {
119     @Override
120     public boolean onCreateActionMode(ActionMode mode, Menu menu) {
121         // inflate contextual menu
122         mode.getMenuInflater().inflate(R.menu.context_menu, menu);
123         return true;
124     }
125
126     @Override
127     public boolean onPrepareActionMode(ActionMode mode, Menu menu) {
128         return false;
129     }
130
131     @Override
132     public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
133         switch (item.getItemId()) {
134             case R.id.menu_delete:
135                 // retrieve selected items and delete them out
136         }
137     }
138 }

```



```

137         SparseBooleanArray selected = laptopListAdapter
138             .getSelectedIds();
139         for (int i = (selected.size() - 1); i >= 0; i--) {
140             if (selected.valueAt(i)) {
141                 Laptop selectedItem = laptopListAdapter
142                     .getItem(selected.keyAt(i));
143                 laptopListAdapter.remove(selectedItem);
144             }
145         }
146         mode.finish(); // Action picked, so close the CAB
147         return true;
148     default:
149         return false;
150     }
151 }
152 }
153
154 @Override
155 public void onDestroyActionMode(ActionMode mode) {
156     // remove selection
157     laptopListAdapter.removeSelection();
158     mActionMode = null;
159 }
160 }
161 }

```

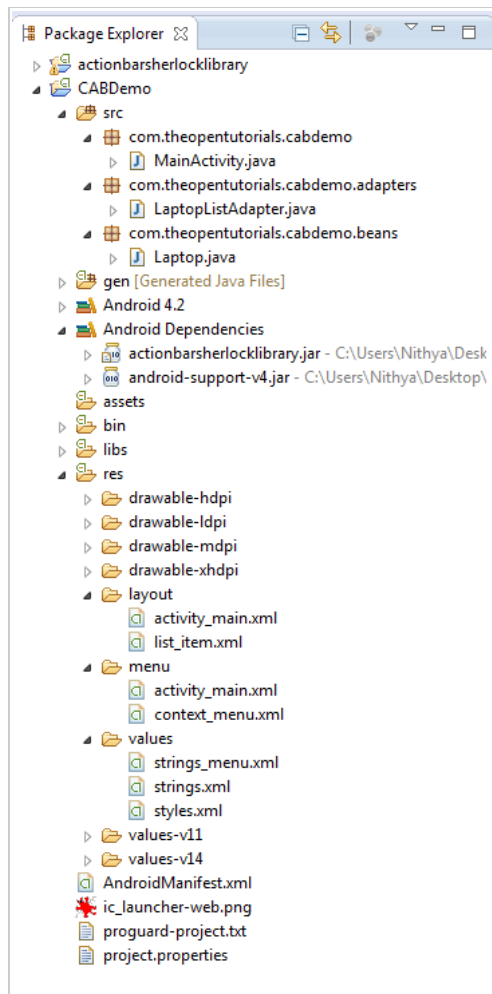
Here, we won't use ListView's built in selection handling. In the case of long click, actionMode activation required switching between different choice modes of ListView like multipleChoice and none, which works buggy. So we have manually added several methods in adapter class for handling item selection.

5. Output

Run your application and you will get the output as shown in the beginning of this tutorial.

6. Folder Structure

The complete folder structure of this project is shown below.



You may also like:

1. [Android: Creating and populating ListView items in XML](#)
2. [Android: Create ListView in XML and populate items using ArrayAdapter](#)
3. [Android: Multiple Selection ListView](#)
4. [Android: ListView in ListActivity](#)
5. [Android: Custom ListView with Image and Text using BaseAdapter](#)

Tags: [android context action bar actionbarsherlock](#), [android contextual action bar with listview](#), [Android Examples](#), [android listem item deletion cab](#), [Android ListView](#), [android listview cab](#), [Android ListView Example](#), [android listview long press context menu](#), [context menu listview android example](#), [how to apply context menu in listview in android](#)

AROUND THE WEB

Son Saves Dad's Retirement with \$396k 'Hidden in Bible'

Moneynews

These 5 Signs Warn You That Cancer Is Starting Inside Your Body.

Newsmax Health

ALSO ON THE OPEN TUTORIALS

How to retrieve all rows from MySQL table using JDBC

3 comments

Rules for Developing EJB 3.x Session Beans

1 comment

WHAT'S THIS?

3 comments

Leave a message...

Best

Community

Share

siddhu • 2 months ago

sir can u please put the download link

3 ^ | v • Reply • Share >

Shaun • a month ago

Works like a charm! Thanks very much for putting this together.

^ | v • Reply • Share >

Nima • 3 months ago

Very buggy, rotate the device and you will be stuck in the Action mode forever :)

^ | v • Reply • Share >

Subscribe

Add Disqus to your site

Views

Pagination in Servlet and JSP - 71,123 views

How to create a simple EJB3 project in Eclipse (JBoss 7.1) - 64,991 views

Android: Custom ListView with Image and Text using ArrayAdapter - 63,450 views

How to create a Servlet with Eclipse and Tomcat - 57,860 views

How to create and consume a simple Web Service using JAX WS - 51,195 views

Recent Posts

Android: Contextual Action Bar for ListView Item Deletion using ActionBarSherlock

Android BadTokenException: Unable to add window, is your activity running

Android: Expandable List View Example

How to retrieve all rows from MySQL table using JDBC

How to retrieve a row from MySQL using JDBC

November 2013

M	T	W	T	F	S	S
			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	
« Jul						

theopentutorials.com/examples/android/listview/android-contextual-action-bar-for-listview-item-deletion-using-actionbarsherlo...

10/10