Android GridView Example

Posted on February 21, 2012
By mkyong

In Android, `GridView` let you arranges components in a two-dimensional scrolling grid. For detail attribute exaplanation, see GridView reference.

In this tutorial, we will show you 2 common `GridView` examples :

1. Normal way, just display text in `GridView` layout.
2. Create a custom adapter to display image and text in `GridView` layout.

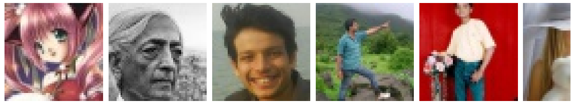
P.S This project is developed in Eclipse 3.7, and tested with Android 2.3.3.

1. Normal GridView example

Display characters from A to Z in `GridView` layout. Quite simple, it should be sef-explanatory.

1.1 Android Layout file – res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gridView1"
```



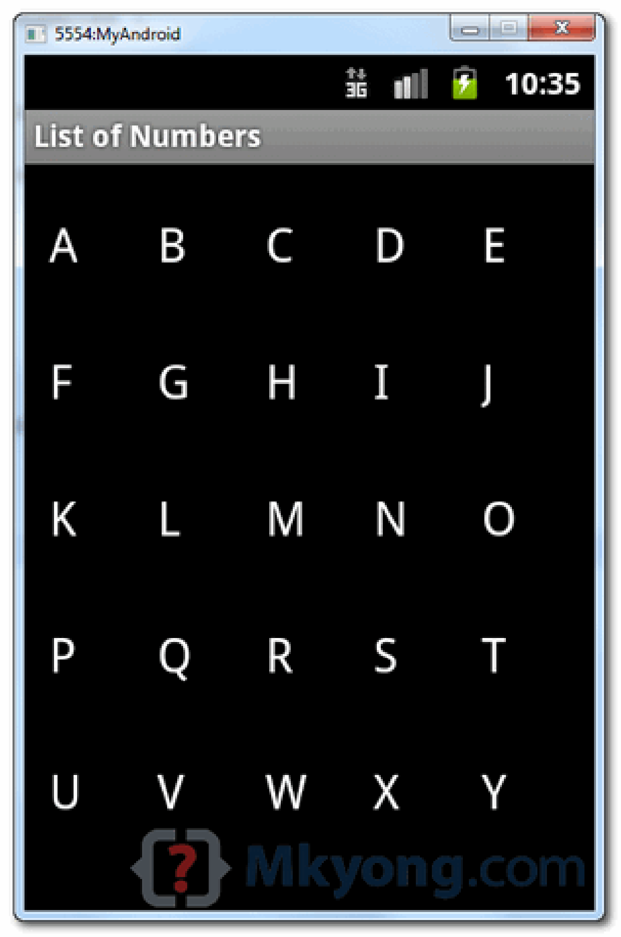
```
    android:numColumns="auto_fit"
    android:gravity="center"
    android:columnWidth="50dp"
    android:stretchMode="columnWidth"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

</GridView>
```

1.2 Activity

```
package com.mkyong.android;

import android.app.Activity;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.GridView;
import android.widget.TextView;
import android.widget.Toast;
import android.view.View;
import android.widget.AdapterView.OnItemClickListener;

public class GridViewActivity extends Activity {

        GridView gridView;

        static final String[] numbers = new String[] {
                        "A", "B", "C", "D", "E",
                        "F", "G", "H", "I", "J",
                        "K", "L", "M", "N", "O",
                        "P", "Q", "R", "S", "T",
                        "U", "V", "W", "X", "Y", "Z"};

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

                setContentView(R.layout.main);
```

```
                gridView = (GridView) findViewById(R.id.gridView1);

                ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
                                android.R.layout.simple_list_item_1, numbers);

                gridView.setAdapter(adapter);

                gridView.setOnItemClickListener(new OnItemClickListener() {
                        public void onItemClick(AdapterView<?> parent, View v,
                                int position, long id) {
                           Toast.makeText(getApplicationContext(),
                                ((TextView) v).getText(), Toast.LENGTH_SHORT).show();
                        }
                });

        }

}
```

1.3 Demo



5554:MyAndroid

10:35

List of Numbers

```
A  B  C  D  E

F  G  H  I  J

K  L  M  N  O

P  Q  R  S  T

U  V  W  X  Y
```

Mkyong.com

2. Custom Adapter example

In this example, extend a `BaseAdapter` to display a group of image and text in `GridView` layout.

2.1 Two Android Layout files

File – res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gridView1"
    android:numColumns="auto_fit"
    android:gravity="center"
    android:columnWidth="100dp"
    android:stretchMode="columnWidth"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

</GridView>
```

File – res/layout/mobile.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="5dp" >

    <ImageView
        android:id="@+id/grid_item_image"
        android:layout_width="50px"
        android:layout_height="50px"
        android:layout_marginRight="10px"
        android:src="@drawable/android_logo" >
    </ImageView>

    <TextView
        android:id="@+id/grid_item_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@+id/label"
        android:layout_marginTop="5px"
        android:textSize="15px" >
    </TextView>
```

```
</LinearLayout>
```

2.2 Custom Adapter

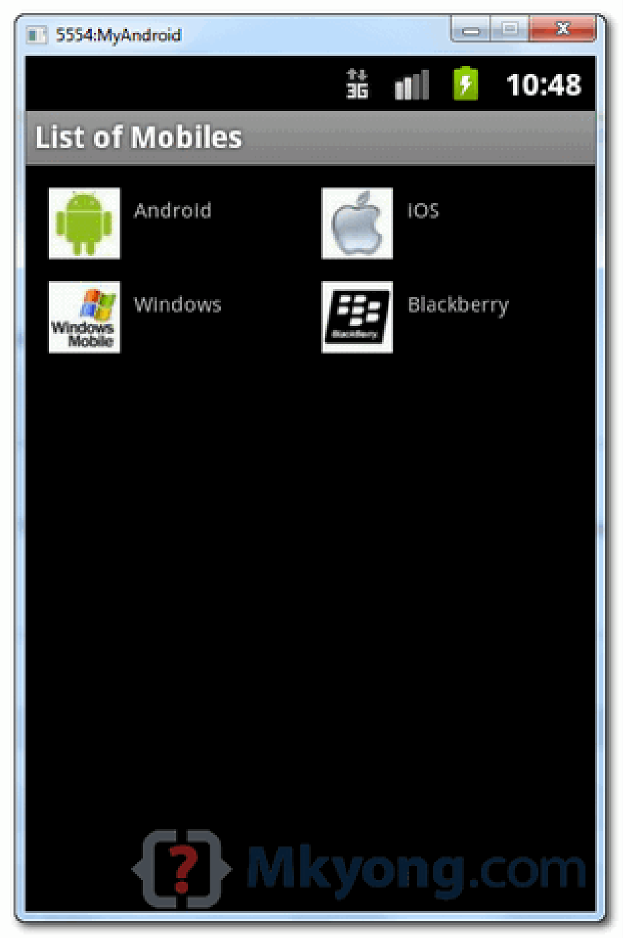
```
package com.mkyong.android.adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import com.mkyong.android.R;

public class ImageAdapter extends BaseAdapter {
        private Context context;
        private final String[] mobileValues;

        public ImageAdapter(Context context, String[] mobileValues) {
                this.context = context;
                this.mobileValues = mobileValues;
        }

        public View getView(int position, View convertView, ViewGroup parent) {

                LayoutInflater inflater = (LayoutInflater) context
                        .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

                View gridView;

                if (convertView == null) {

                        gridView = new View(context);

                        // get layout from mobile.xml
                        gridView = inflater.inflate(R.layout.mobile, null);
                        // set value into textview
                        TextView textView = (TextView) gridView
                                        .findViewById(R.id.grid_item_label);
                        textView.setText(mobileValues[position]);

                        // set image based on selected text
                        ImageView imageView = (ImageView) gridView
                                        .findViewById(R.id.grid_item_image);

                        String mobile = mobileValues[position];

                        if (mobile.equals("Windows")) {
                                imageView.setImageResource(R.drawable.windows_logo);
                        } else if (mobile.equals("iOS")) {
                                imageView.setImageResource(R.drawable.ios_logo);
                        } else if (mobile.equals("Blackberry")) {
                                imageView.setImageResource(R.drawable.blackberry_logo);
                        } else {
                                imageView.setImageResource(R.drawable.android_logo);
                        }

                } else {
                        gridView = (View) convertView;
                }

                return gridView;
        }

        @Override
        public int getCount() {
                return mobileValues.length;
        }

        @Override
        public Object getItem(int position) {
                return null;
        }

        @Override
        public long getItemId(int position) {
                return 0;
        }

}
```

2.3 Activity

```
package com.mkyong.android;

import com.mkyong.android.adapter.ImageAdapter;
import android.app.Activity;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.GridView;
import android.widget.TextView;
import android.widget.Toast;
import android.view.View;
import android.widget.AdapterView.OnItemClickListener;

public class GridViewActivity extends Activity {

        GridView gridView;

        static final String[] MOBILE_OS = new String[] {
                "Android", "iOS","Windows", "Blackberry" };

        @Override
        public void onCreate(Bundle savedInstanceState) {

                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);

                gridView = (GridView) findViewById(R.id.gridView1);

                gridView.setAdapter(new ImageAdapter(this, MOBILE_OS));

                gridView.setOnItemClickListener(new OnItemClickListener() {
                        public void onItemClick(AdapterView<?> parent, View v,
                                        int position, long id) {
```

```
                                Toast.makeText(
                                   getApplicationContext(),
                                   ((TextView) v.findViewById(R.id.grid_item_label))
                                   .getText(), Toast.LENGTH_SHORT).show();

                        }
                });

        }

}
```

2.4 Demo



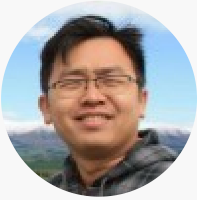
Download Source Code

Download it – Android-GridView-Example.zip (21 KB)

References

1. GridView Javadoc
2. Android GridView example
3. Android ListView example

Tags : android gridview layout



mkyong

Founder of Mkyong.com, love Java and open source stuffs. Follow him on Twitter, or befriend him on Facebook or Google Plus.

Related Posts

- Android TabLayout example
- Android TableLayout example
- Android RelativeLayout example
- Android LinearLayout example
- Android Tutorial

Popular Posts

- Top 8 Java People You Should Know
- Top 20 Java Websites
- Top 5 Free Java eBooks
- Top 10 Java Regular Expression Examples
- Top 5 Open Source Q&A Systems

You might also like following tutorials :

Zipacz

January 6, 2013 at 6:42 am

Hi, I still find this tutorial very helpful, but “It doesn’t work!”. I mean I made a mistake somewhere. The tutorial itself is fully functional. I have a method which takes some strings from my SQL database:

```
  public String[] getDataBase() {
        String[] columns = new String[]{KEY_ONE, KEY_TWO, KEY_THREE};
        Cursor c = myDataBase.query(DB_TABLE, columns, null, null, null, null, null
        String[] result = new String[3*c.getCount()];
        int i = 0;

        int iOne = c.getColumnIndex(KEY_ONE);
        int iTwo = c.getColumnIndex(KEY_TWO);
        int iThree = c.getColumnIndex(KEY_THREE);
            for (c.moveToFirst(); !c.isAfterLast(); c.moveToNext()){
                result[i] =  c.getString(iOne);
                result[i+1] =  c.getString(iTwo);
                result[i+2] =  c.getString(iThree);
                i=i+3;
            }
    return result;
    }
```

Then I take result array in another class

```
String[] data = info.getDataBase();
```

And put it into adapter

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list
       gv.setAdapter(adapter);
```

, where gv is existing gridView

```
GridView gv = (GridView) findViewById(R.id.gview);
```

When I run all the app just with

```
gv.setAdapter(adapter);
```

commented, everything works fine. I also tried to show data[1] + data[2] + data[3] in TextView and everything works fine, but when I uncomment

```
gv.setAdapter(adapter);
```

application “unfortunately stop working”.

Can You help me?

Reply

Zipacz

January 6, 2013 at 5:27 am

Thank You a lot. I am a total beginner and this helped me pretty much. Thanks

Reply

Umesha

December 17, 2012 at 6:32 pm

This helps me a lot..Thanks for the tutorial

Reply

Anselmo

December 13, 2012 at 6:35 am

Thank you very much for the tutorial. Greetings from Mexico.

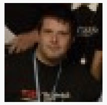
Reply

Fun With Puzzles

October 23, 2012 at 2:52 pm

Thanks very much for this easy to understand tutorial.

Reply



Roberto Allende

October 23, 2012 at 11:45 am

Hello

Thank you very much, great post!, source code worked very well.

Reply

Jason

October 13, 2012 at 3:57 pm

Hi and thank you for your tutorials, I am learning programming and would love for you to give a tutorial on using GridView to show a "month view" calendar with the ability to display a image/icon on certain user selected days?
Once again thank you I have found your tutorials very helpful and hope that you will create this tutorial.
jason

Reply

pavan

September 25, 2012 at 1:56 pm

hi watch this

pavanhd.blogspot.in

Reply

khan

July 31, 2012 at 2:38 pm

owsum….

http://www.sportshunter.eu

Reply

Bob

July 29, 2012 at 5:48 am

I'm trying to learn Android programming, and I think that the GridView will help me with my first application.
I tried copying your "Normal Gridview Example" that should display the letters A thru Z, but after copying your main.xml code and Activity code into my project, the Eclipse notifications place a red-X on the line for:
public class GridViewActivity extends Activity {
The red-X description says:
The public type GridViewActivity must be defined in its own file.
The name of the file that the Activity code is in is called "GridviewActivity.java".
Do you know what I'm doing wrong?

Reply

Steve

July 29, 2012 at 5:46 am

Hi exelent tutorial but i am struggling in centering the grid view horizontally, can you provide some tips? thanks

Reply

iqra

July 28, 2012 at 6:22 pm

i want that when i click on string array value mean text so it goes to next activity please help me

Reply

iqra

July 28, 2012 at 6:21 pm

i want when i click on string array element means text so it goes to next activity.is it possible please help me

Reply

cuongtc

July 1, 2012 at 11:49 am

thank a lot

Reply

elma

June 23, 2012 at 3:20 pm

I want to set a background to the rows of the grid, is it possible?
May you guide how I can do it?
Thanks

Reply

Arvind

June 23, 2012 at 3:44 am

Thank you. Lucid and simple as always.

Reply

raju

June 4, 2012 at 11:01 pm

can you please tell me,how to modify your activity class and imageadapter for getting images+title text from remote server database?

Reply

Carlos

March 13, 2012 at 8:51 am

Muchas gracias por el tutorial, me ha sido de ayuda!
Thanks for the tutorial! =D

Reply

Neel

February 22, 2012 at 4:40 pm

How to get the name of selected item into string rather then using

```
Toast.makeText(getApplicationContext(),
((TextView) v.findViewById(R.id.grid_item_label))
.getText(), Toast.LENGTH_SHORT).show();
```

i want to store the result in 1 string
..
Thank You

Reply

I'd love your thoughts

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

Note

To post source code in comment, uses `<pre lang="language"></pre>` tag, for examples :

- `<pre lang="java"> Java source code here </pre>`
- `<pre lang="xml"> XML here </pre>`

Post Comment

□ Notify me of followup comments via e-mail

All Available Tutorials

Java Core Technologies :
Java I/O, Java RegEx, Java XML, Java JSON, JDBC, Java Misc

J2EE Frameworks :
Hibernate, JSF 2.0, Spring Core, Spring MVC, Spring Security, Apache Wicket, Struts 1.x, Struts 2.x

Web Service :
JAX-WS (SOAP), JAX-RS (REST)

Build Tools :
Maven, Archiva

Unit Test Frameworks :
jUnit, TestNG

Others :
Android, Google App Engine, jQuery, Java MongoDB, Quartz Scheduler

Favorites Links

Android Developer
Google App Engine using Java
DZone - Fresh Links
Official Java EE 5 Tutorial
Official Java EE 6 Tutorial
Spring 2.5.x documentation
Spring 3.1.x documentation
Hibernate core documentation
Java SE 6.0 API documentation
Java EE 6.0 API documentation
Java Secure Socket Extension (JSSE) Reference Guide
JSP home page
JSF home page
Eclipse IDE for Java developer
Struts 1.3 documentation
Struts 2.2 documentation
Maven home page
Maven central repository Search
Java.Net Maven repository
Ant home page

Friends & Links

Java Training
JavaScript Training
Java Code Geeks
PHP Tutorials
TenthOfMarch
Find Free Icons
Web Security Blog
Web Development

About Us

Mkyong.com is a weblog dedicated to Java/J2EE developers and Web Developers. We constantly publish useful tricks, tutorials on J2EE or web development.

All examples are simple, easy to read, and full source code available, and of course well tested in our development environment.

We're Social

1. Twitter - Follow Me
2. Facebook - Like Me
3. Google Plus - Add Me
4. RSS - Subscribe Me

JAX-WS Official Website

JAX-RS Official Website (Jersey)

MongoDB Official Website