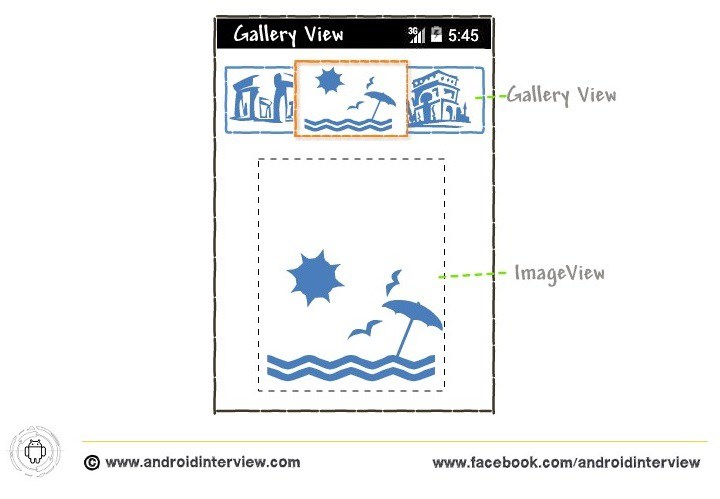
**Android Gallery View – Displaying a List of Images**

Posted by [R.K.Kanojia (Android App Developer)](http://www.androidinterview.com/author/rkkanojia7/)

Android Gallery View :

In your application if you want display a series of images to the user, you can make use of the Gallery.The Gallery is a view that shows items (such as images) in a center-locked, horizontal scrolling list.

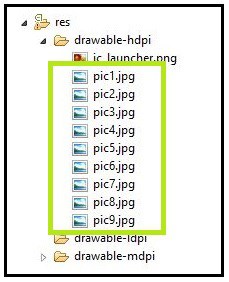
[](http://i2.wp.com/www.androidinterview.com/wp-content/uploads/2014/08/android-Gallery-View-example-2.jpg)

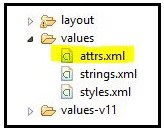
How to Create Android Gallery View:

* Gallery is used to show Views in a horizontal list, and user will select a view , User selected view will be shown in center of the Horizontal list
* The items of Gallery are get from an Adapter, just like ListView, in which ListView items are get from an Adapter.
* We need to make an Adapter class that extends BaseAdapter class and override  getView() method.
* getView() method known as automatically for all items of Gallery

Solution Stuff :

1. For Android Gallery View Example, assume you have some images stored in the res/drawable-mdpi folder of your project (see Figure).

[](http://i2.wp.com/www.androidinterview.com/wp-content/uploads/2014/08/android-Gallery-View-example.jpg)  
2. Create an XML file named attrs.xml and store it in the res/values folder (see Figure).

[](http://i0.wp.com/www.androidinterview.com/wp-content/uploads/2014/08/android-Gallery-View-example1.jpg)3. Add  this content to the attrs.xml file:

*File: res/values/attrs.xml*

Java



|  |  |
| --- | --- |
| 1  2  3  4  5 | <resources>  <declare-styleable name="MyGallery">  <attr name="android:galleryItemBackground" />  </declare-styleable>  </resources> |

4. Open “**res/layout/activity\_main.xml**” file and put this xml code it into file:

To use the Gallery, add the <Gallery> element in your UI, such as the activity\_main.xml file:

*File : res/layout/activity\_main.xml*

activity\_main.xml

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 | <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"      xmlns:tools="http://schemas.android.com/tools"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      tools:context="{relativePackage}.${activityClass}"      android:orientation="vertical" >        <Gallery          android:id="@+id/gallery1"          android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content" />        <ImageView          android:id="@+id/image1"          android:layout\_width="215dp"          android:layout\_height="315dp"          android:layout\_gravity="center\_horizontal"          android:background="#cfcfcf"          android:paddingTop="5dp"          android:paddingBottom="5dp"          android:paddingLeft="10dp"          android:paddingRight="10dp"          android:src="@drawable/pic1" />  </LinearLayout> |

5. Open “**MainActivity.java**” file and add following JAVA code.

*File : src/package-name/MainActivity.java*

MainActivity.java

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 | package androidinterview.com.androidgalleryview;    import android.app.Activity;  import android.content.Context;  import android.content.res.TypedArray;  import android.os.Bundle;  import android.view.View;  import android.view.ViewGroup;  import android.widget.AdapterView;  import android.widget.AdapterView.OnItemClickListener;  import android.widget.BaseAdapter;  import android.widget.Gallery;  import android.widget.ImageView;  import android.widget.Toast;    *@SuppressWarnings*("deprecation")  public class MainActivity extends Activity {    //the images to display  Integer[] imageIDs = {  R.drawable.pic1,  R.drawable.pic2,  R.drawable.pic3,  R.drawable.pic4,  R.drawable.pic5,  R.drawable.pic6,  R.drawable.pic7  };  *@Override*  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);    // Note that Gallery view is deprecated in Android 4.1---  Gallery gallery = (Gallery) findViewById(R.id.gallery1);  gallery.setAdapter(new ImageAdapter(this));  gallery.setOnItemClickListener(new OnItemClickListener() {  public void onItemClick(AdapterView<?> parent, View v, int position,long id)  {  Toast.makeText(getBaseContext(),"pic" + (position + 1) + " selected",  Toast.LENGTH\_SHORT).show();  // display the images selected  ImageView imageView = (ImageView) findViewById(R.id.image1);  imageView.setImageResource(imageIDs[position]);  }  });  }    public class ImageAdapter extends BaseAdapter {  private Context context;  private int itemBackground;  public ImageAdapter(Context c)  {  context = c;  // sets a grey background; wraps around the images  TypedArray a =obtainStyledAttributes(R.styleable.MyGallery);  itemBackground = a.getResourceId(R.styleable.MyGallery\_android\_galleryItemBackground, 0);  a.recycle();  }  // returns the number of images  public int getCount() {  return imageIDs.length;  }  // returns the ID of an item  public Object getItem(int position) {  return position;  }  // returns the ID of an item  public long getItemId(int position) {  return position;  }  // returns an ImageView view  public View getView(int position, View convertView, ViewGroup parent) {  ImageView imageView = new ImageView(context);  imageView.setImageResource(imageIDs[position]);  imageView.setLayoutParams(new Gallery.LayoutParams(100, 100));  imageView.setBackgroundResource(itemBackground);  return imageView;  }  }  } |

In Android Gallery View , you need create the ImageAdapter class which extends the BaseAdapter class.Which will bind to the Gallery view with a series of ImageView views. The BaseAdapter class will work as a bridge between an AdapterView and also the data source that feeds data into it.

For the ImageAdapter class, you implemented the following methods:

* getCount()
* getItem()
* getItemId()
* getView()

When a picture in the Gallery view is selected, the selected image’s position (0 for the first image, 1 for the second image, and so on) is displayed and the image is displayed in the ImageView.

 Output of  Android  Gallery View Example in emulator :

[](http://i1.wp.com/www.androidinterview.com/wp-content/uploads/2014/08/android-Gallery-View-example-3.jpg)

Download Android Gallery View example Source code :

|  |
| --- |
| Icon  **Android Gallery View example Source code**5.69 MB  [**Android Gallery View example Download**](http://www.androidinterview.com/android-gallery-view-example-displaying-a-list-of-images/) |