

Module 3 : UI Design and Data storage

List Views

List views are views that enable you to display a long list of items. In Android, there are two types of list views: ListView and SpinnerView. Both are useful for displaying long lists of items.

Listview View

The ListView displays a list of items in a vertically scrolling list. The following demonstrates how to display a list of items using the ListView.

```
packagenet.learn2develop.BasicViews5;
import android.app.Activity;
import android.os.Bundle;
import android.app.ListActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.Toast;
public class MainActivity extends ListActivity {
    String[] presidents = { "Dwight D. Eisenhower", "John F. Kennedy", "Lyndon B.
    Johnson", "Richard Nixon", "Gerald Ford", "Jimmy Carter", "Ronald Reagan",
    "George H. W. Bush", "Bill Clinton", "George W. Bush", "Barack Obama" };
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    { super.onCreate(savedInstanceState);
    // setContentView(R.layout.main);
    setListAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
    presidents)); }
    public void onItemClick( ListView parent, View v, int position, long id)
    { Toast.makeText(this, "You have selected " + presidents[position],
    Toast.LENGTH_SHORT).show(); } }
```



FIGURE 4-15

Customizing the ListView

The ListView is a versatile control that you can further customize. The following Try It Out shows how you can allow multiple items in the ListView to be selected and how you can enable filtering support.

MainActivity.java file:

```

/**Calledwhentheactivityisfirstcreated.*/
@Override
publicvoidonCreate(Bundle savedInstanceState)
{ super.onCreate(savedInstanceState);
  setContentView(R.layout.main);
  ListView listView = getListView(); //listView.setChoiceMode(0);
  //CHOICE_MODE_NONE //listView.setChoiceMode(1);
  //CHOICE_MODE_SINGLE listView.setChoiceMode(2);
  //CHOICE_MODE_MULTIPLE listView.setTextFilterEnabled(true);
  setListAdapter(new ArrayAdapter<String>(this,
  android.R.layout.simple_list_item_checked, presidents)); }
publicvoidonListItemClick( ListView parent, View v, int position, long id)
{ //--toggle the check displayed next to the item--parent.setItemChecked(position,
parent.isItemChecked(position));
  Toast.makeText(this, "Youhaveselectd"+presidents[position],
  Toast.LENGTH_SHORT).show(); }

```



FIGURE 4-16

Using Image Views To Display Pictures

For displaying images, you can use the `ImageView`, `Gallery`, `ImageSwitcher`, and `GridView` views.

Gallery And Imageview Views

The `Gallery` is a view that shows items (such as images) in a center-locked, horizontal scrolling list. Figure 5-1 shows how the `Gallery` view looks when it is displaying some images.



FIGURE 5-1

```

package net.learn2develop.Gallery;
import android.app.Activity;
import android.os.Bundle;
import android.content.Context;
import android.content.res.TypedArray;
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;
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 };
    /**Called when the activity is first created.*/ @Override
    public void onCreate(Bundle savedInstanceState)
    { super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    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(getApplicationContext(), "pic" + (position + 1) + " selected",
    Toast.LENGTH_SHORT).show(); } }); }
    public class ImageAdapter extends BaseAdapter
    { private Context context; private int itemBackground;
    public ImageAdapter(Context c)
    { context = c; //---setting the style--TypedArray a =
    obtainStyledAttributes(R.styleable.Gallery1);
    itemBackground = a.getResourceId(
    R.styleable.Gallery1_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.setScaleType(ImageView.ScaleType.FIT_XY);
    imageView.setLayoutParams(new Gallery.LayoutParams(150, 120));
    imageView.setBackgroundResource(itemBackground);

```

```
return imageView; } } }
```



FIGURE 5-5



FIGURE 5-6

To display the selected image in the `ImageView`, add the following statements in bold to the `MainActivity.java` file:

```
/**Calledwhen the activity is first created.*/  
@Override  
public void onCreate(Bundle savedInstanceState)  
{  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
    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]);  
        }  
    });  
}
```



FIGURE 5-7

Imageswitcher

The previous section demonstrated how to use the Gallery view together with an `ImageView` to display a series of thumbnail images so that when one is selected, the selected image is displayed in the `ImageView`. However, sometimes you don't want an image to appear abruptly when the user selects it in the Gallery view — you might, for example, want to apply some animation to the image when it transits from one image to another. In this case, you need to use the `ImageSwitcher` together with the Gallery view. The following Try It Out shows you how.

```
packagenet.learn2develop.ImageSwitcher;
import android.app.Activity;
import android.os.Bundle;
import android.content.Context;
import android.content.res.TypedArray;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewGroup.LayoutParams;
import android.view.animation.AnimationUtils;
import android.widget.BaseAdapter;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Gallery;
import android.widget.ViewSwitcher.ViewFactory;
```

```

import android.widget.ImageSwitcher;
import android.widget.ImageView;
public class MainActivity extends Activity implements ViewFactory
{
    //---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 };
    private ImageSwitcher imageSwitcher;
    /**Called when the activity is first created.*/
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        imageSwitcher = (ImageSwitcher) findViewById(R.id.switcher1);
        imageSwitcher.setFactory(this);
        imageSwitcher.setInAnimation(AnimationUtils.loadAnimation(this,
        android.R.anim.fade_in));
        imageSwitcher.setOutAnimation(AnimationUtils.loadAnimation(this,
        android.R.anim.fade_out));
        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)
            {
                imageSwitcher.setImageResource(imageIDs[position]);
            }
        });
    }
    public View makeView()
    {
        ImageView imageView = new ImageView(this);
        imageView.setBackgroundColor(0xFF000000);
        imageView.setScaleType(ImageView.ScaleType.FIT_CENTER);
        imageView.setLayoutParams(new ImageSwitcher.LayoutParams(
        LayoutParams.FILL_PARENT, LayoutParams.FILL_PARENT));
        return imageView;
    }
    public class ImageAdapter extends BaseAdapter {
        private Context context;
        private int
        itemBackground;
        public ImageAdapter(Context c)

```

```

    { context = c;
    //---setting the style--TypedArray a = obtainStyledAttributes(R.styleable.Gallery1);
    itemBackground          =          a.getResourceId(
    R.styleable.Gallery1_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; }
    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.setScaleType(ImageView.ScaleType.FIT_XY);
    imageView.setLayoutParams(new Gallery.LayoutParams(150, 120));
    imageView.setBackgroundResource(itemBackground); return imageView; } } }

```



FIGURE 5-8

GridView

The GridView shows items in a two-dimensional scrolling grid. You can use the GridView together with an ImageView to display a series of images. The following Try It Out demonstrates how.


```

packagenet.learn2develop.Grid;
import android.app.Activity;
import android.os.Bundle;
import android.content.Context;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ImageView;
import android.widget.Toast;

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 };
    /**Called when the activity is first created.*/
    @Override
    public void onCreate(Bundle savedInstanceState)
    { super.onCreate(savedInstanceState); setContentView(R.layout.main);
    GridView gridView = (GridView) findViewById(R.id.gridview);
    gridView.setAdapter(new ImageAdapter(this));
    gridView.setOnItemClickListener(new OnItemClickListener()
    { public void onItemClick(AdapterView<?> parent, View v, int position, long id) {
    Toast.makeText(getApplicationContext(), "pic" + (position + 1) + " selected",
    Toast.LENGTH_SHORT).show(); } }); }
    public class ImageAdapter extends BaseAdapter
    { private Context context;
    public ImageAdapter(Context c)
    { context = c; }
    //---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; if (convertView == null)
{ imageView = new ImageView(context);
imageView.setLayoutParams(new GridView.LayoutParams(85, 85));
imageView.setScaleType( ImageView.ScaleType.CENTER_CROP);
imageView.setPadding(5, 5, 5, 5); }
else { imageView = (ImageView) convertView; }
imageView.setImageResource(imageIDs[position]);
return imageView; } } }

```



FIGURE 5-10