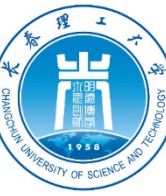


# 移动应用开发

张昕

zhangxin@cust.edu.cn

2022



# Review 1

---

## □ UI Components

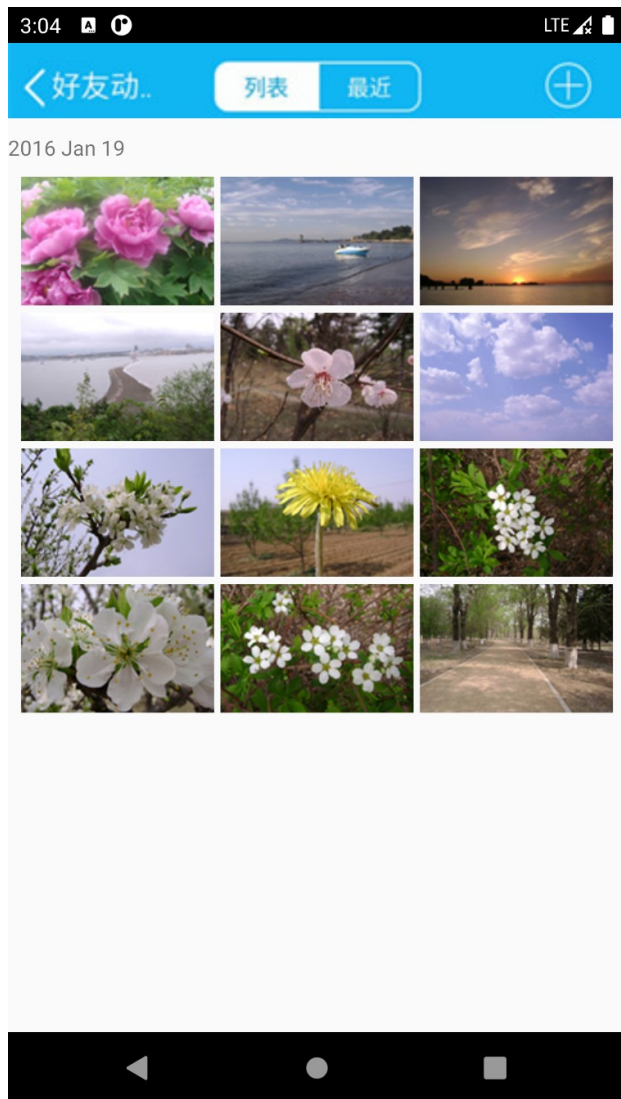
### □ GridView

- When using GridView, Adapter class (it is in fact an interface) is often employed to offer the relevant data, the implementing classes are:
  - **ArrayAdapter:** wraps multiple values of array into several list items and only displays one-line text
  - **SimpleAdapter:** wraps multiple values of List set into several list items. It is powerful and easy-to-customized
  - **SimpleCursorAdapter:** must align the Cursor attribute to component ID. It could display the content in database as list

### □ ListView

- Load data in ListView
  - **Solution 1:** In res/values folder, you could establish a xml file containing string array to load entries in ListView
  - **Solution 2:** (*Similar to GridView*)
    - Create Adapter object. ArrayAdapter object is often used to load list of text.
    - Link the created adapter to the ListView object with setAdapter() method.

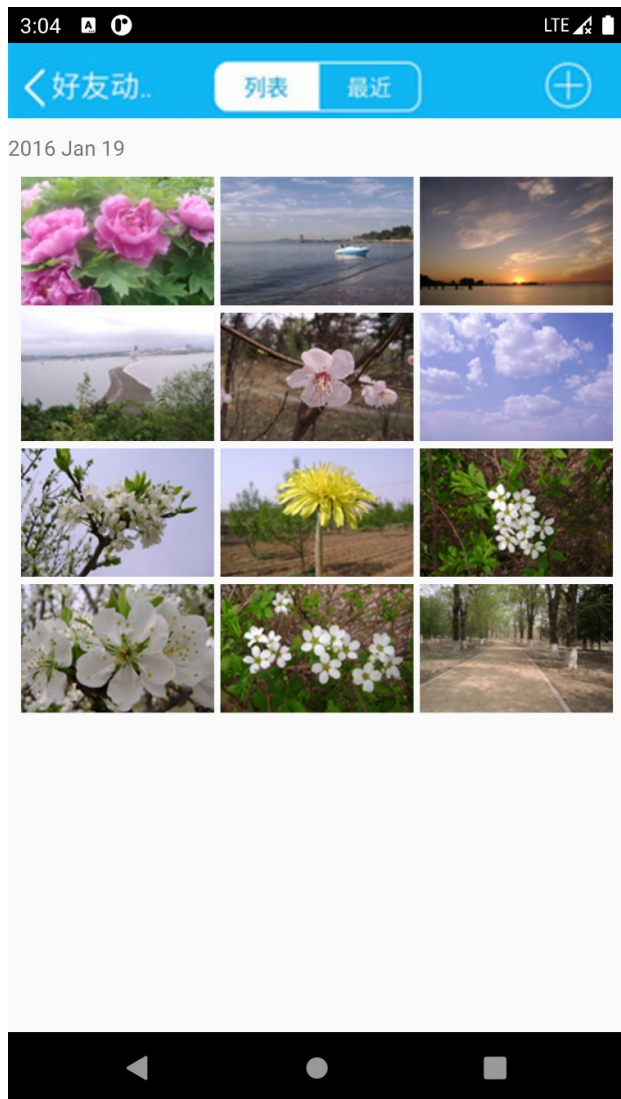
# Small Demo



```
<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"
    android:orientation="vertical"
    tools:context="com.example.qqalbum.MainActivity">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@mipmap/qxixiang"
        android:scaleType="centerCrop"
        android:adjustViewBounds="true"
        android:maxLength="1440dp" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:text="2016 Jan 19" />
    <GridView
        android:id="@+id/gridview1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:columnWidth="120dp"
        android:gravity="center"
        android:numColumns="auto_fit"
        android:stretchMode="spacingWidthUniform"
        android:verticalSpacing="5dp">
    </GridView>
</LinearLayout>
```



# Small Demo

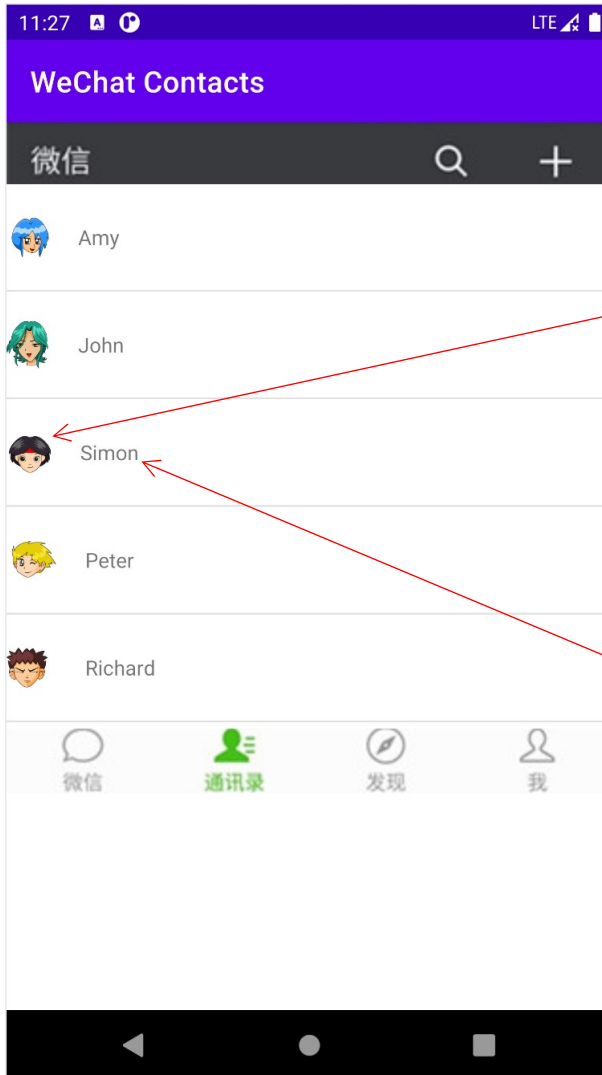


```
public class MainActivity extends Activity {
    private Integer[] picture = {R.mipmap.img01,R.mipmap.img02,R.mipmap.img03,R.mipmap.img04,
        R.mipmap.img05,R.mipmap.img06,R.mipmap.img07,R.mipmap.img08,R.mipmap.img09,
        R.mipmap.img10,R.mipmap.img11,R.mipmap.img12,};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        GridView gridView = (GridView)findViewById(R.id.gridview1);
        gridView.setAdapter(new ImageAdapter(this));
    }

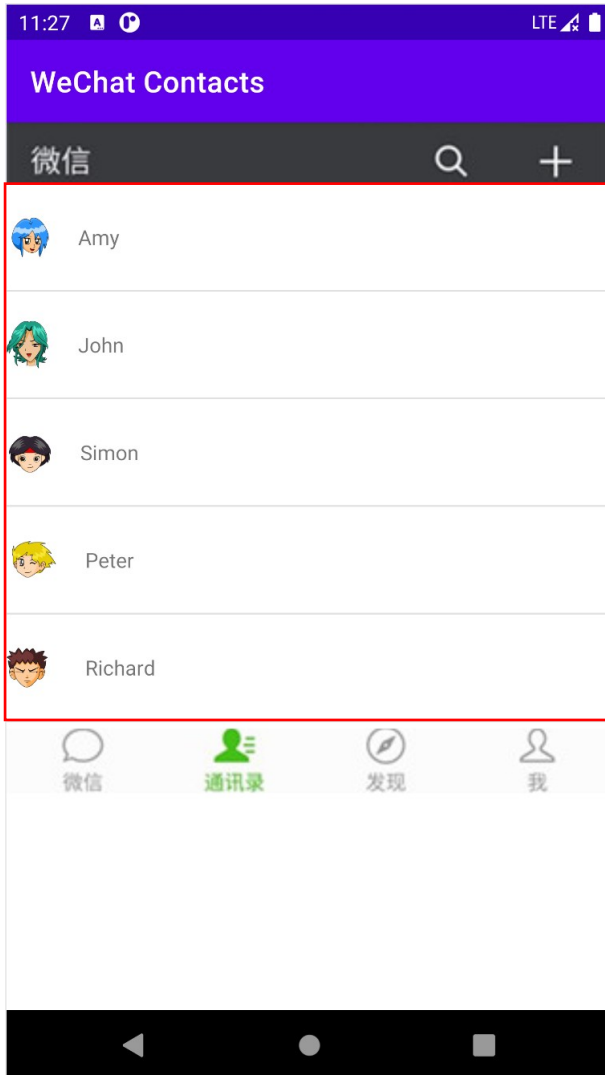
    public class ImageAdapter extends BaseAdapter{
        private Context mContext;
        public ImageAdapter(Context c){
            mContext = c;
        }
        public int getCount(){
            return picture.length;
        }
        public Object getItem(int position){
            return null;
        }
        public long getItemId(int position){
            return 0;
        }
        public View getView(int position, View convertView, ViewGroup parent){
            ImageView imageView;
            if(convertView == null){
                imageView = new ImageView(mContext);
                imageView.setLayoutParams(new ViewGroup.LayoutParams(450,300));
                imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
            }
            else{
                imageView = (ImageView)convertView;
            }
            imageView.setImageResource(picture[position]);
            return imageView;
        }
    }
}
```

# Small Demo



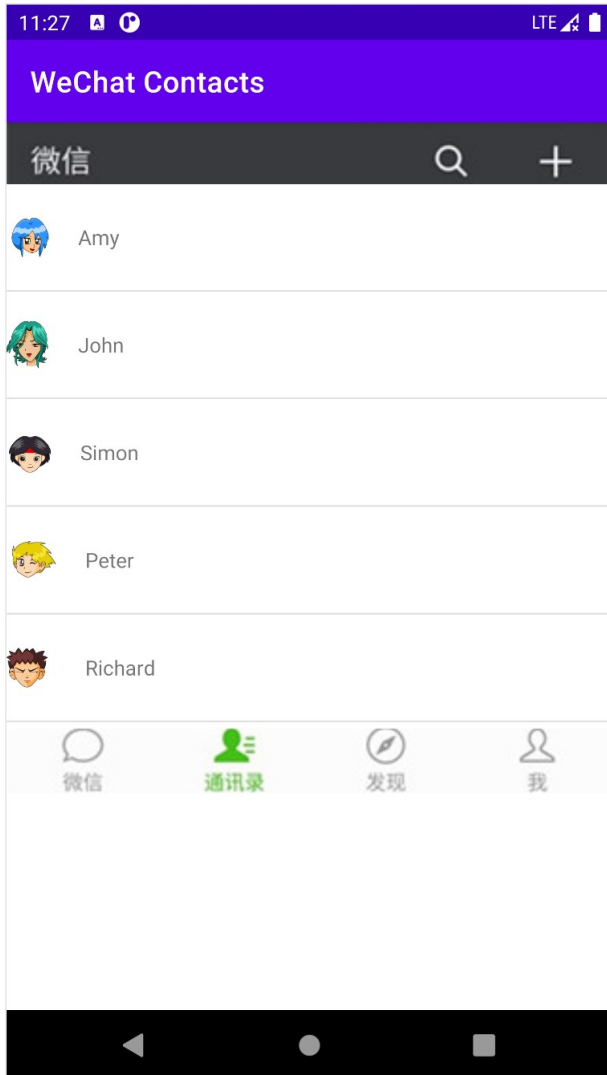
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:id="@+id/image"
        android:paddingRight="10dp"
        android:paddingTop="20dp"
        android:paddingBottom="20dp"
        android:adjustViewBounds="true"
        android:maxWidth="72dp"
        android:maxHeight="72dp"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:layout_gravity="center"
        android:id="@+id/title"
    />
</LinearLayout>
```

# Small Demo



```
<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"
    android:orientation="vertical"
    tools:context="com.example.wechatcontacts.MainActivity">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@mipmap/wei_top"
        android:scaleType="centerCrop"/>
    <ListView
        android:id="@+id/listview"
        android:layout_width="match_parent"
        android:layout_height="370dp">
    </ListView>
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@mipmap/wei_down"
        android:scaleType="centerCrop"/>
</LinearLayout>
```

# Small Demo



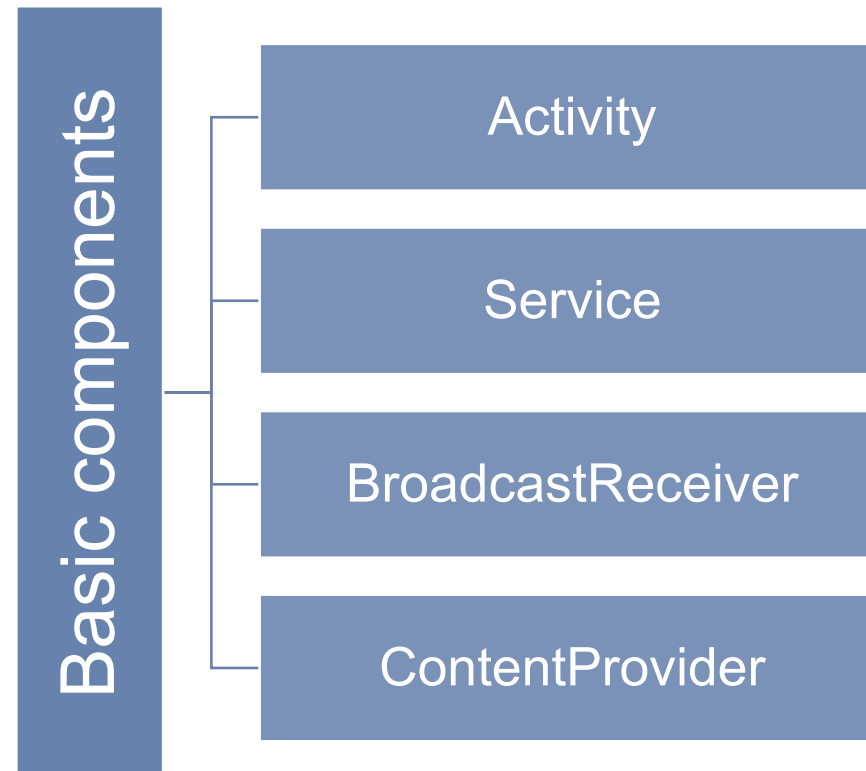
```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ListView listview = (ListView) findViewById(R.id.listview);
        int[] imageId = new int[]{R.mipmap.img01, R.mipmap.img02, R.mipmap.img03,
            R.mipmap.img04, R.mipmap.img05, R.mipmap.img06,
            R.mipmap.img07, R.mipmap.img08, R.mipmap.img09,
        };
        String[] title = new String[]{"Amy", "John", "Simon", "Peter", "Richard",
            "Lee", "Eva", "Catherine", "Mozart"};
        List<Map<String, Object>> listItems = new ArrayList<Map<String, Object>>();

        for (int i = 0; i < imageId.length; i++) {
            Map<String, Object> map = new HashMap<String, Object>();
            map.put("image", imageId[i]);
            map.put("Name", title[i]);
            listItems.add(map);
        }
        SimpleAdapter adapter = new SimpleAdapter(this, listItems,
            R.layout.main, new String[]{"Name", "image"}, new int[]{
                R.id.title, R.id.image});
        listview.setAdapter(adapter);
        listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Map<String, Object> map = (Map<String, Object>) parent.getItemAtPosition(position);
                Toast.makeText(MainActivity.this, map.get("Name").
                    toString(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

# Review 2

## □ Basic Components

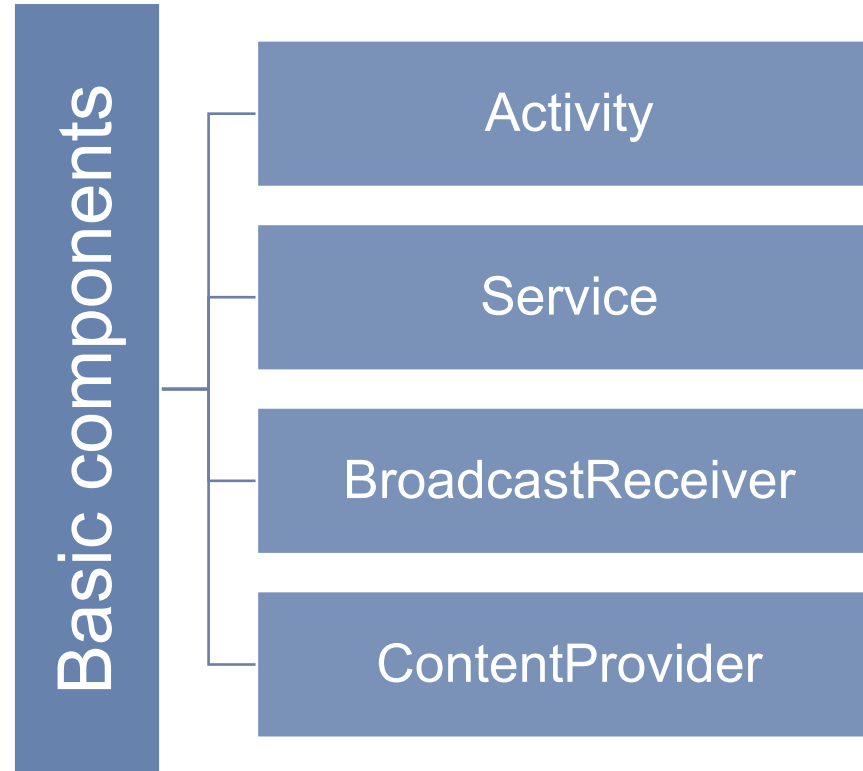
- as the basic building blocks to develop the Android application capability of interacting with users





# Android Basic Components

---





# Activity

---

- **One Activity reflects one screen frame that interacts with users**
  - There could be multiple activities within one app
- **All the activities constitute an Activity Stack**
  - The only active activity is on the top of the stack

# Activity

## ■ Four states of Activity

### □ Running

- Current active activity, on the stack top
- Visible to users, has focus

### □ Pause

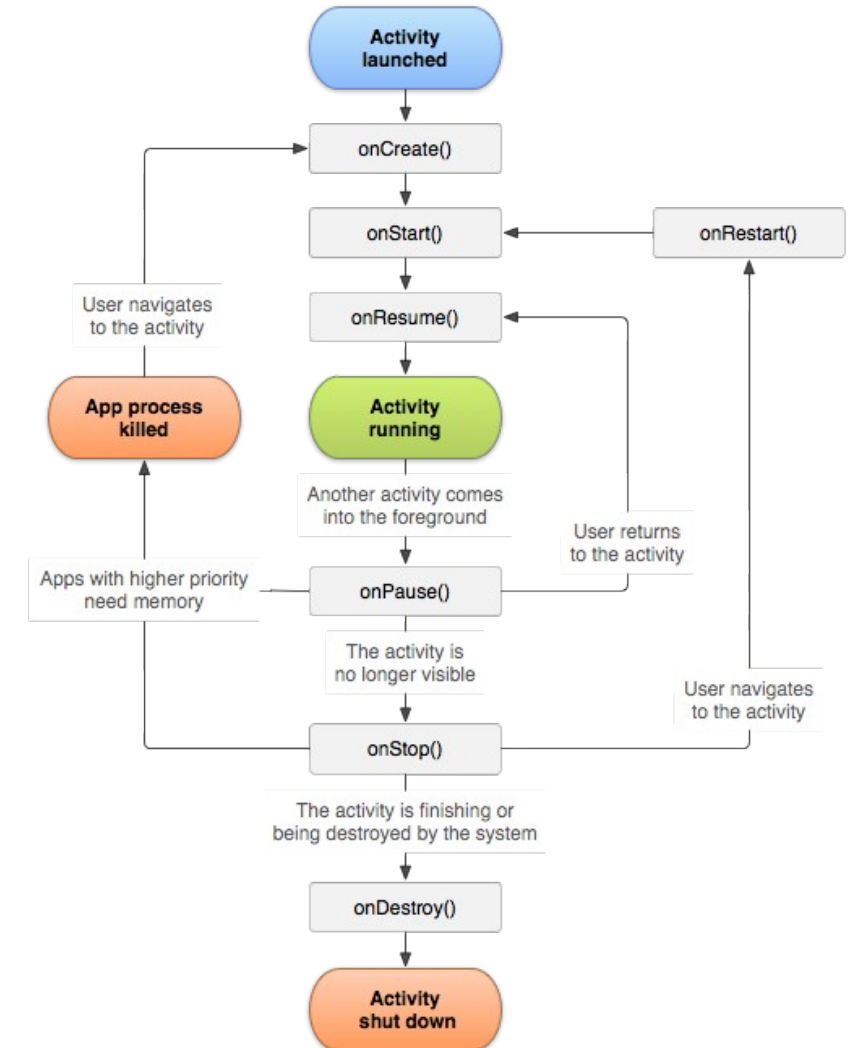
- Loses focus, still visible
- Would be killed if low memory

### □ Stop

- Overtaken by other activity, invisible
- Still keep status and information, kill if low memory

### □ Destroyed

- Activity ends, all the processes are disposed

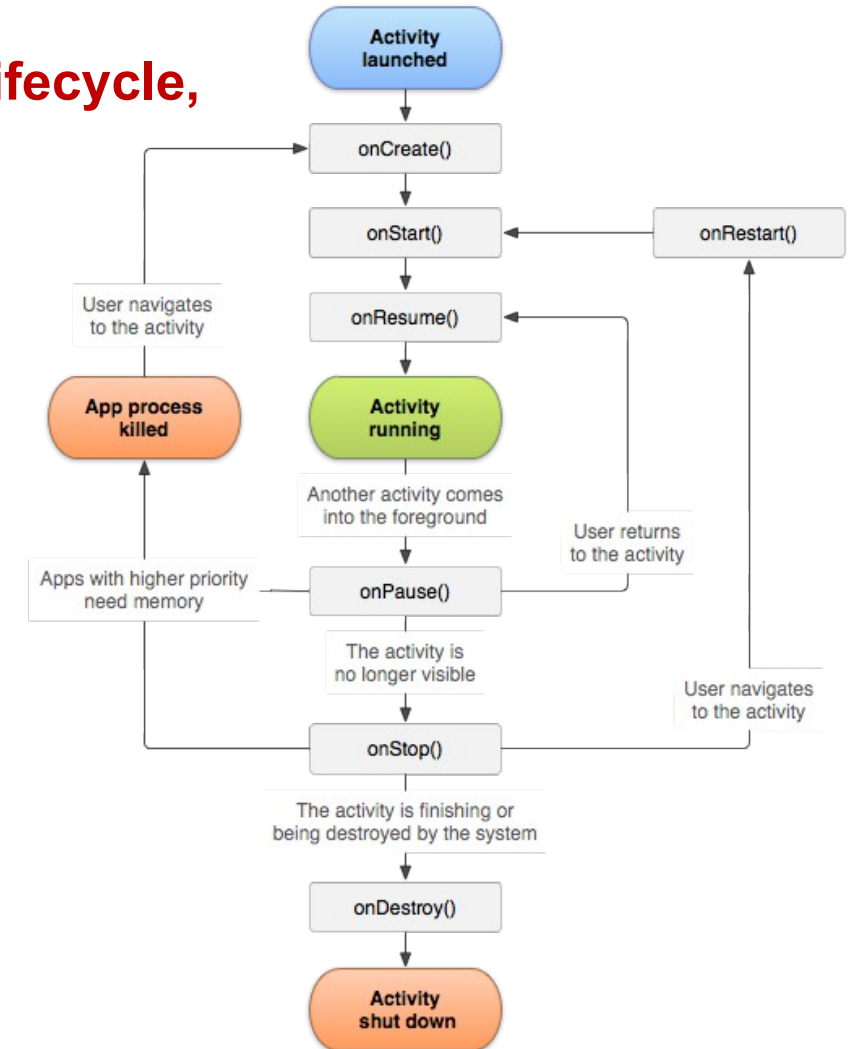


# Activity

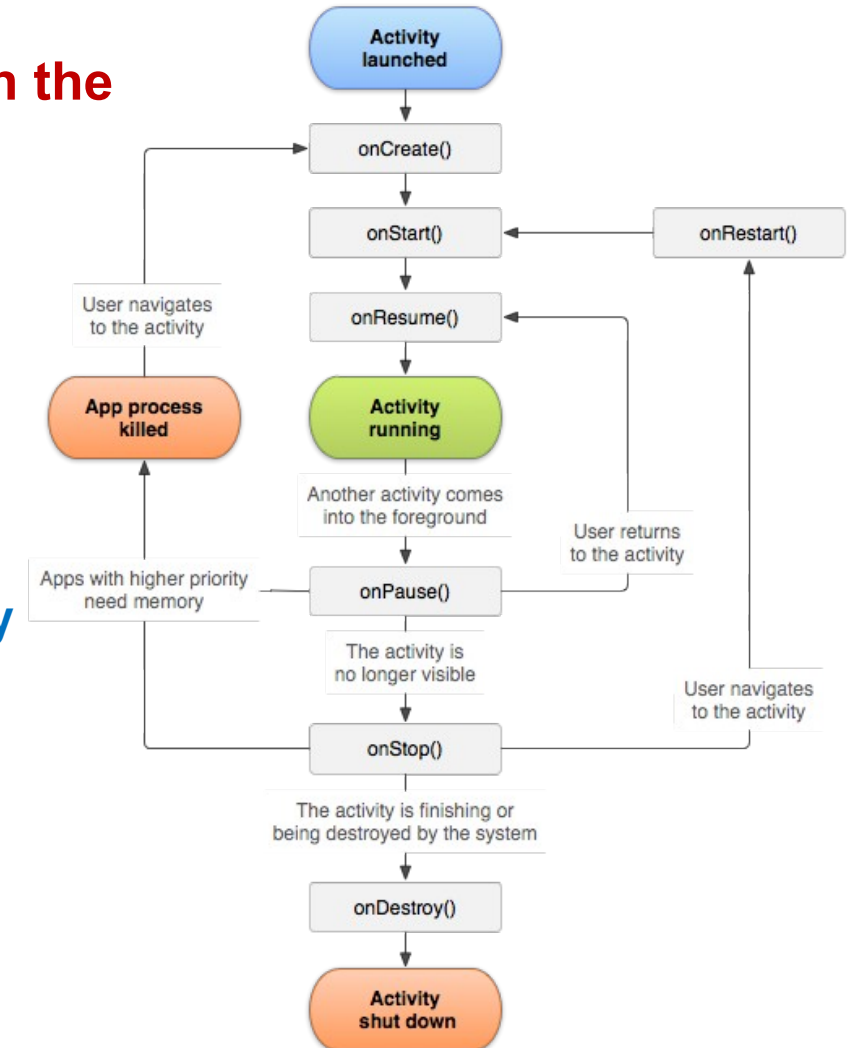
- To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks:

- onCreate()
- onStart()
- onResume()
- onPause()
- onStop()
- onDestroy()

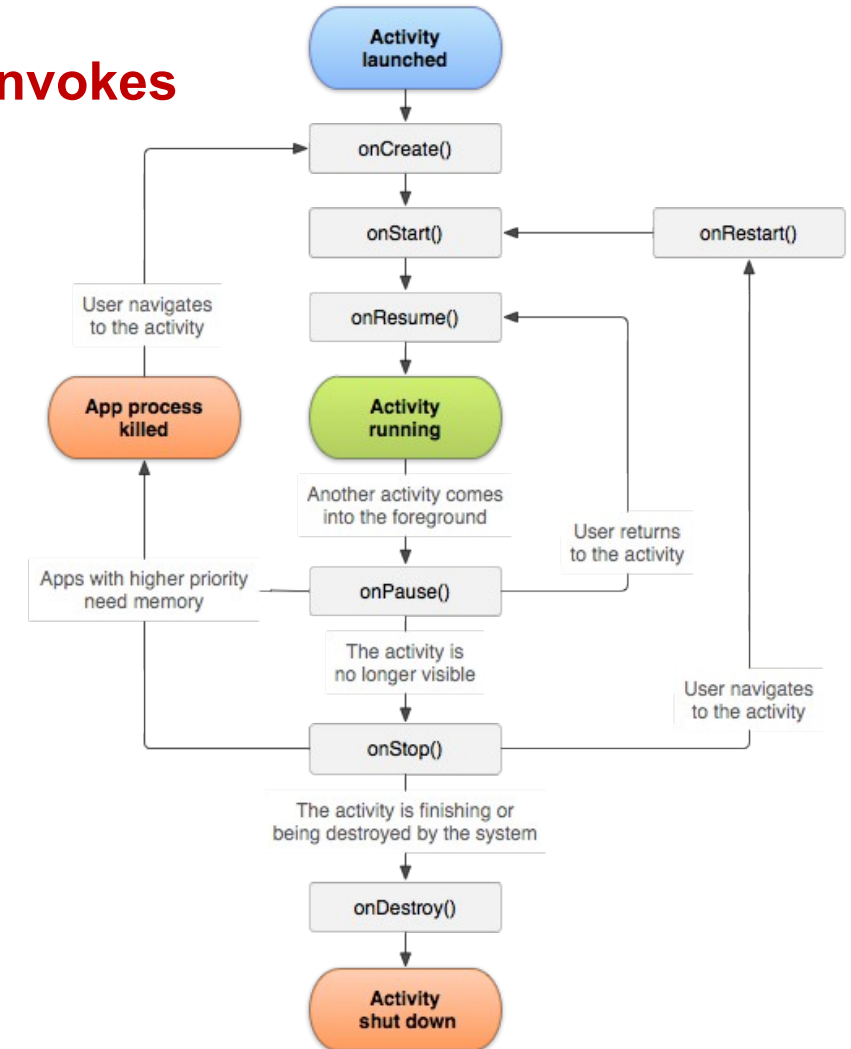
- The system invokes each of these callbacks as an activity enters a new state.



## onCreate() callback



## onStart() callback



# onResume() callback

---

- **When the activity enters Resumed state, it comes to foreground, and the system invokes the onResume() callback.**
  - In Resumed state, the app interacts with the user and stays in the state until something happens to take focus away from the app.
    - e.g., when receiving a phone call, the user's navigating to another activity, or the device screen's turning off.
  - When the activity moves to the resumed state, any lifecycle-aware component tied to the activity's lifecycle will receive the ON\_RESUME event.
  - When an interruptive event occurs, the activity enters the Paused state, and the system invokes the onPause() callback.
    - If the activity returns to the Resumed state from the Paused state, the system once again calls onResume() method.

# onPause() callback

- **The system calls this method as the first indication that the user is leaving the activity (though not being destroyed); it indicates the activity is no longer in the foreground.**
  - When the activity moves to the paused state, any lifecycle-aware component tied to the activity's lifecycle will receive the ON\_PAUSE event.
    - This is where the lifecycle components can stop any functionality that does not need to run while the component is not in the foreground, such as stopping a camera preview.
  - onPause() method could release system resources, handle to sensors (i.e., GPS), or any resources affecting battery life while the activity is paused and the user doesn't need them.
    - a Paused activity may still be fully visible if in multi-window mode.
  - Completion of the onPause() method does not mean that the activity leaves the Paused state.
    - activity remains in this state until either the activity resumes or becomes completely invisible to users.
    - If the activity resumes, the system once again invokes the onResume() callback. If the activity returns from the Paused state to the Resumed state, the system keeps the Activity instance resident in memory, recalling that instance when the system invokes onResume().



# onStop() callback

---

- **When the activity is no longer visible to the user, it has entered the Stopped state, and the system invokes the onStop() callback.**
  - When the activity moves to the stopped state, any lifecycle-aware component tied to the activity's lifecycle will receive the ON\_STOP event.
    - This is where the lifecycle components can stop any functionality that does not need to run while the component is not visible on the screen.
  - In the onStop() method, the app should release or adjust resources that are not needed while the app is not visible to the user.
  - When the activity enters the Stopped state, the Activity object is kept resident in memory: It maintains all state and member information, but is not attached to the window manager.
  - From the Stopped state, the activity either comes back to interact with the user, or the activity is finished running and goes away.
    - If the activity comes back, the system invokes onRestart().
    - If the Activity is finished running, the system calls onDestroy().

# onDestroy() callback

---

- **onDestroy() is called before the activity is destroyed.**
  - The system invokes this callback either because:
    - the activity is finishing (due to the user completely dismissing the activity or due to finish() being called on the activity), or
    - the system is temporarily destroying the activity due to a configuration change (such as device rotation or multi-window mode)
  - When the activity moves to the destroyed state, any lifecycle-aware component tied to the activity's lifecycle will receive the ON\_DESTROY event.
    - This is where the lifecycle components can clean up anything it needs to before the Activity is destroyed.
  - The onDestroy() callback should release all resources that have not yet been released by earlier callbacks such as onStop().

# Create an Activity

---

## ■ Steps

- New an Activity through extending `android.app.Activity` or sometimes extending sub-class of Activity
  - e.g., `ListActivity`, `TabActivity`
- Override callback method(s), `onCreate()` is obligated

Android Studio also supports to create Activity by menu wizard

# Configure an Activity

- **AndroidManifest.xml provides configuration of Activity**

- Within `<application></application>`

- Insert an Activity by adding `<activity></activity>` tags

```
<activity  
    android:name="the class implementing Activity"  
    android:label="label text"  
    android:theme="the theme to be used"  
    ...  
>  
</activity>
```

The package path should be configured carefully to make the corresponding class found correctly

# Start an Activity

- **When an android app only has one Activity, configure the Activity with `AndroidManifest.xml` and set it as the app entry**
- **When there are multiple Activities, use `startActivity()` to start the correct Activity**

`public void startActivity(Intent intent)`

- Intent is the entry argument
- Intent is the specific communication means
  - An Activity utilize Intent to perform its own intention
  - When creating an Intent object, the target Activity should be declared

```
Intent intent = new Intent(MainActivity.this, DetailActivity.class);  
startActivity(intent);
```

# Shut down an Activity

- **Call finish() method provided by Activity class**

```
public void finish()
```

- **Call the method when necessary**

- **e.g., within an event handling method**

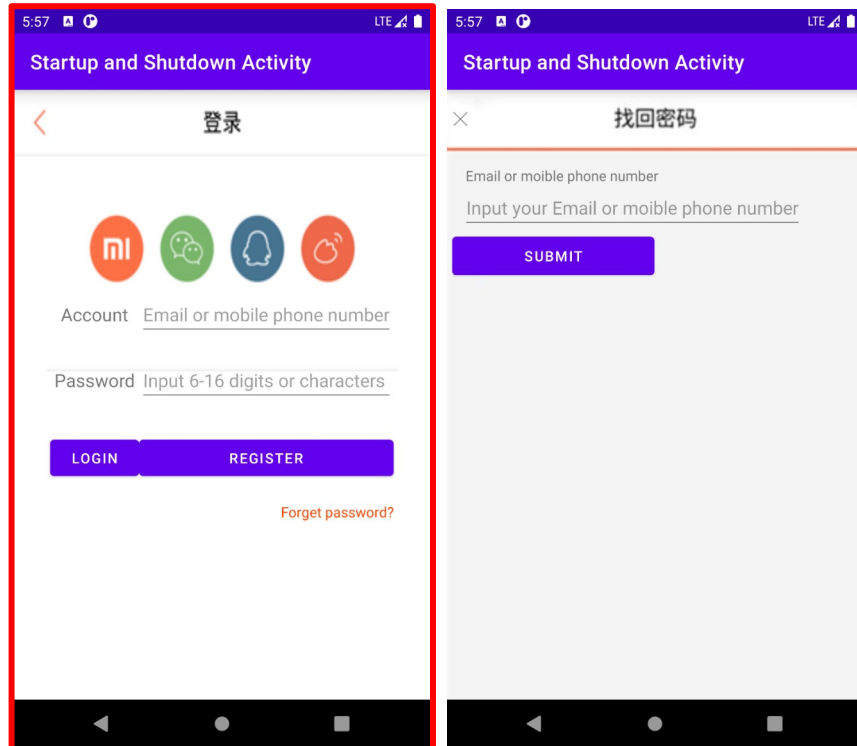
```
Button button1 = (Button)findViewById(R.id.button1);  
button1.setOnClickListener(new View.OnClickListener(){  
    @override  
    public void onClick(View v){  
        finish();  
    }  
});
```

If the current Activity is not the main Activity, finish() method would make the program return to the Activity calling the current Activity. Otherwise, the app will return to main screen.



# Demo

## Modify activity\_main.xml and set TableLayout 4 TableRow



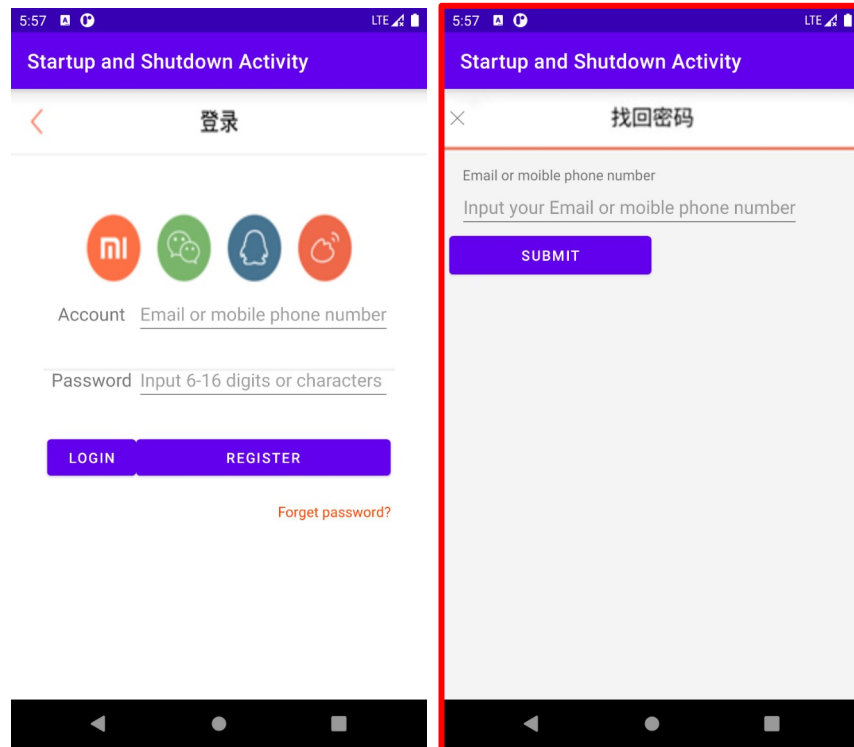
```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/biaoge"
    tools:context="com.example.startupandshutdownactivity.MainActivity">
    <TableRow
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="200dp"
        android:gravity="center_horizontal">
        <TextView/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="18sp"
            android:text="Account"
            android:gravity="center_horizontal"/>
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Email or mobile phone number"/>
        <TextView/>
    </TableRow>
    <TableRow
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="20dp"
        android:gravity="center_horizontal">
        <TextView/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="18sp"
            android:text="Password"
            android:gravity="center_horizontal"/>
        <EditText
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:hint="Input 6-16 digits or characters"/>
        <TextView/>
    </TableRow>
```

```
<TableRow
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"
    android:layout_marginTop="30dp">
    <TextView/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#FF8247"
        android:text="Register"/>
    <TextView/>
</TableRow>
<TableRow
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"
    android:paddingTop="20dp">
    <TextView/>
    <TextView/>
    <TextView
        android:id="@+id/wang_mima"
        android:text="Forget password?"
        android:textColor="#FF4500"
        android:gravity="right"/>
    <TextView/>
</TableRow>
</TableLayout>
```

# Demo

Create activity\_password.xml and set vertical LinearLayout



Create an Activity named PasswordActivity whose layout is activity\_password.xml

```
<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"
    android:background="@drawable/beijing"
    android:orientation="vertical"
    android:padding="5dp"
    tools:context="com.example.startupandshutdownactivity.PasswordActivity">
    <ImageButton
        android:id="@+id/close"
        android:layout_width="15dp"
        android:layout_height="15dp"
        android:layout_marginTop="15dp"
        android:background="#0000"
        android:scaleType="fitXY"
        android:src="@drawable/guanbi" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="13dp"
        android:layout_marginTop="40dp"
        android:text="Email or mobile phone number" />
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:hint="Input your Email or mobile phone number" />
    <Button
        android:layout_width="200dp"
        android:layout_height="50dp"
        android:background="#FF8247"
        android:text="Submit" />
</LinearLayout>
```



# Demo

- Create PasswordActivity.java through extending Activity and set the corresponding layout as activity\_password.xml

```
public class PasswordActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_password);//set layout  
        ImageButton close = (ImageButton) findViewById(R.id.close);  
        close.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                finish();  
            }  
        });  
    }  
}
```

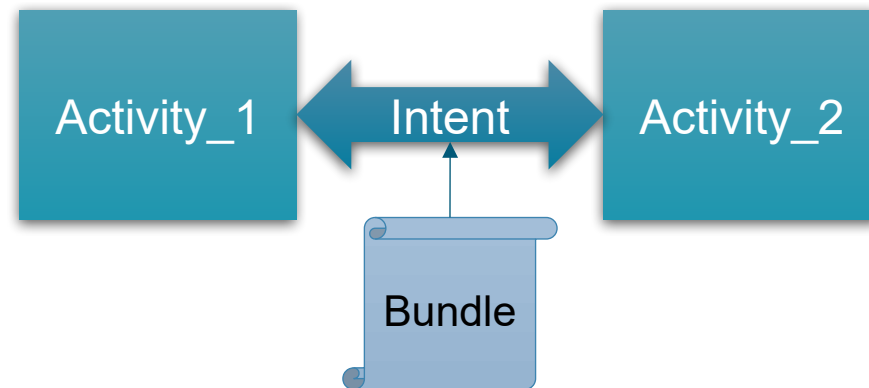
# Demo

- ❑ Adjust MainActivity and make it extend AppCompatActivity.
- ❑ Within onCreate(), add an event listener.
- ❑ Override onClick() and create the Intent object corresponding to PasswordActivity, then call startActivity() to invoke PasswordActivity

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        TextView password = (TextView)findViewById(R.id.wang_mima);  
        password.setOnClickListener(new View.OnClickListener(){  
            @Override  
            public void onClick(View v) {  
                Intent intent = new Intent(MainActivity.this, PasswordActivity.class);  
                startActivity(intent);  
            }  
        });  
    }  
}
```

# Multiple Activities

- **Exchange data with Bundle among Activities**
  - Intent is used to transfer data from one Activity to another one
  - Bundle object preserves the data that could be saved to Intent by calling putExtras()



Bundle is a combination of key-value pairs

The contained data type could be boolean, byte, int, long, float, double, String, Objects, or arrays of the types



# Demo

8:37

收货地址管理

吉林省长春市南关区

卫星路

7186号

精神小伙

13800138000

130022

SAVE

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.activityexchangedata.MainActivity">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:scaleType="centerCrop"
        android:src="@drawable/top" />
    <EditText
        android:id="@+id/et_site1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Province"
        android:paddingTop="50dp" />
    <EditText
        android:id="@+id/et_site2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/et_site1"
        android:hint="Street" />
    <EditText
        android:id="@+id/et_site3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/et_site2"
        android:hint="Number" />
    <EditText
        android:id="@+id/et_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/et_site3"
        android:hint="Name" />
```

```
<EditText
    android:id="@+id/et_phone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/et_name"
    android:hint="Mobile number" />
<EditText
    android:id="@+id/et_email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/et_phone"
    android:hint="Postcode" />
<Button
    android:id="@+id/btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="10dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="10dp"
    android:background="#FF5000"
    android:text="Save" />
</RelativeLayout>
```



# Demo

8:46 LTE

← 收货地址管理

吉林省长春市南关区

卫星路

7186号

Name

13800138000

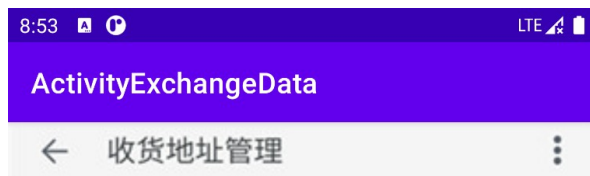
130022

Please complement the whole address

SAVE

```
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button btn = (Button)findViewById(R.id.btn);
        btn.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                String site1 = ((EditText)findViewById(R.id.et_site1)).getText().toString();
                String site2 = ((EditText)findViewById(R.id.et_site2)).getText().toString();
                String site3 = ((EditText)findViewById(R.id.et_site3)).getText().toString();
                String name = ((EditText)findViewById(R.id.et_name)).getText().toString();
                String phone = ((EditText)findViewById(R.id.et_phone)).getText().toString();
                String email = ((EditText)findViewById(R.id.et_email)).getText().toString();
                if(!"".equals(site1)&&!"".equals(site2)&&!"".equals(site3)&&
                    !"".equals(name)&&!"".equals(phone)&&!"".equals(email)){
                    Intent intent = new Intent(MainActivity.this, AddressActivity.class);
                    Bundle bundle = new Bundle();
                    bundle.putCharSequence("name", name);
                    bundle.putCharSequence("phone", phone);
                    bundle.putCharSequence("site1", site1);
                    bundle.putCharSequence("site2", site2);
                    bundle.putCharSequence("site3", site3);
                    intent.putExtras(bundle);
                    startActivity(intent);
                }else{
                    Toast.makeText(MainActivity.this, "Please complement the whole address",
                        Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

# Demo



精神小伙  
吉林省长春市南关区卫星路7186号

13800138000



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.activityexchangedata.AddressActivity">
    <ImageView
        android:id="@+id/top"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:scaleType="centerCrop"
        android:src="@drawable/top"/>
    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/top"/>
    <TextView
        android:id="@+id/phone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/site"
        android:layout_alignParentRight="true"
        android:layout_marginRight="30dp"/>
    <TextView
        android:id="@+id/site"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"/>
</RelativeLayout>
```

# Demo



```
public class AddressActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_address);  
        Intent intent = getIntent();  
        Bundle bundle = intent.getExtras();  
        TextView name = (TextView)findViewById(R.id.name);  
        name.setText(bundle.getString("name"));  
        TextView phone = (TextView)findViewById(R.id.phone);  
        phone.setText(bundle.getString("phone"));  
        TextView site = (TextView)findViewById(R.id.site);  
        site.setText(bundle.getString("site1")  
            +bundle.getString("site2")+bundle.getString("site3"));  
    }  
}
```



# Call another Activity to return result

- **From one Activity call another one, the latter completes the tasks and returns the results to the former**
  - **With Intent and Bundle**
  - **Call startActivityForResult() method to invoke another Activity**
    - **When the second Activity finishes and completes tasks, the result is return to the previous Activity**

```
public void startActivityForResult(Intent intent, int requestCode)
```
    - **requestCode is used to invoke an Activity and determined by developers**
    - **the returned result is received through overriding onActivityResult()**



# Demo



<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"
android:background="@drawable/beijing"
android:orientation="vertical"
tools:context="com.example.selectavatar.MainActivity">
```

<ImageView

```
android:id="@+id/imageView"
android:layout_width="100dp"
android:layout_height="100dp"
android:layout_gravity="center_horizontal"
android:src="@drawable/touxiang"/>
```

<Button

```
android:id="@+id/btn"
android:layout_width="wrap_content"
android:layout_height="40dp"
android:layout_gravity="center_horizontal"
android:text="Select avatar"/>
```

</LinearLayout>

# Demo



```
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button button = (Button)findViewById(R.id.btn);
        button.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, HeadActivity.class);
                startActivityForResult(intent, 0x11);
            }
        });
    }

    @Override
    //judge the request code and result code
    //if the code is correct, then the data package is received and display the image
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode == 0x11 && resultCode == 0x11){
            Bundle bundle = data.getExtras();
            int imageId = bundle.getInt("imageId");
            ImageView iv = (ImageView)findViewById(R.id.imageView);
            iv.setImageResource(imageId);
        }
    }
}
```

# Demo

2:35   LTE  



```
<RelativeLayout
    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="com.example.selectavatar.HeadActivity">
    <GridView
        android:id="@+id/gridView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="10dp"
        android:horizontalSpacing="3dp"
        android:verticalSpacing="3dp"
        android:numColumns="4">
    </GridView>
</RelativeLayout>
```





# Demo

2:35 [status icons] LTE [signal strength]



```
public class HeadActivity extends Activity {
    public int[] imageId = new int[]{R.drawable.touxiang1, R.drawable.touxiang2,
        R.drawable.touxiang3, R.drawable.touxiang4, R.drawable.touxiang5};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_head);
        GridView gridView = (GridView)findViewById(R.id.gridView);//get the View to display images
        BaseAdapter adapter = new BaseAdapter() {
            @Override
            public int getCount() { //get the number of images
                return imageId.length;
            }
            @Override
            public Object getItem(int position) { //get the current selected image
                return position;
            }
            @Override
            public long getItemId(int position) { //get ID of the current select image
                return position;
            }
            @Override
            public View getView(int position, View convertView, ViewGroup parent) {
                ImageView imageview; //declare ImageView object
                if(convertView == null){
                    imageview = new ImageView(HeadActivity.this); //initialize ImageView object
                    imageview.setAdjustViewBounds(true);
                    imageview.setMaxWidth(158);
                    imageview.setMaxHeight(150);
                    imageview.setPadding(5,5,5,5);
                }else{
                    imageview = (ImageView)convertView;
                }
                imageview.setImageResource(imageId[position]); //load images into ImageView
                return imageview; //return ImageView object containing images
            }
        };
        gridView.setAdapter(adapter);
        gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Intent intent = getIntent(); //get the Intent for the further exchange
                Bundle bundle = new Bundle();//declare Bundle object for create data package
                bundle.putInt("imageId", imageId[position]); //put data within the Bundle object
                intent.putExtras(bundle); //save data package in Intent object
                setResult(0x11, intent); //return request code to the Activity
                finish(); //shut down the current Activity
            }
        });
    }
}
```



# Demo



```
public class HeadActivity extends Activity {
    public int[] imageId = new int[]{R.drawable.touxiang1, R.drawable.touxiang2,
        R.drawable.touxiang3, R.drawable.touxiang4, R.drawable.touxiang5};

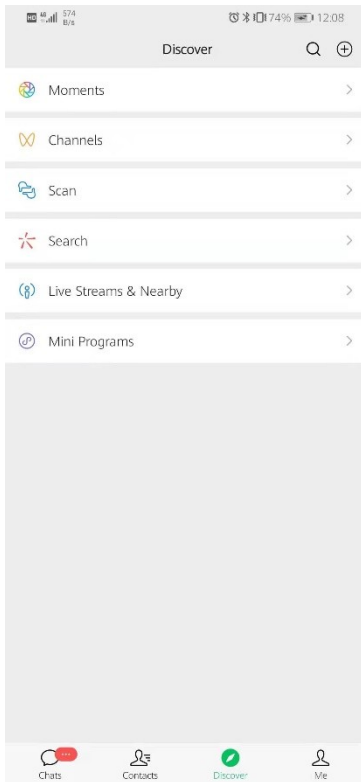
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_head);
        GridView gridView = (GridView)findViewById(R.id.gridView);//get the View to display images
        BaseAdapter adapter = new BaseAdapter() {
            @Override
            public int getCount() { //get the number of images
                return imageId.length;
            }
        };
    }
}
```

```
@Override
//judge the request code and result code
//if the code is correct, then the data package is received and display the image
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode == 0x11 && resultCode == 0x11){
        Bundle bundle = data.getExtras();
        int imageId = bundle.getInt("imageId");
        ImageView iv = (ImageView)findViewById(R.id.imageView);
        iv.setImageResource(imageId);
    }
}
```

```
return imageView; //return ImageView object containing images
};
gridview.setAdapter(adapter);
gridview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Intent intent = getIntent(); //get the Intent for the further exchange
        Bundle bundle = new Bundle();//declare Bundle object for create data package
        bundle.putInt("imageId", imageId[position]); //put data within the Bundle object
        intent.putExtras(bundle); //save data package in Intent object
        setResult(0x11, intent); //return request code to the Activity
        finish(); //shut down the current Activity
    }
});
}
```

# Fragment

- **Similar to Activity, describe some acts and UI within in one Activity**
  - **Multiple Fragment could build up multiple UI panels within one Activity**
  - **The Fragment could be re-used in multiple Activities**



The current UI is corresponding to one Activity.

There are four function, i.e., Chats, Contacts, Discover, Me are four function interfaces, and each of the interfaces is one Fragment. The Fragments could switch to each other.

# Life cycle of Fragment

---

- **One Fragment is embedded into one Activity**
  - Fragment lifecycle is limited by the located Activity
    - When Activity is paused, the contained Fragment is also paused.
    - When Activity is destroyed, the contained Fragment is also destroyed.
    - When Activity is running (in Resumed status), Fragment could be manipulated.

# Create Fragment

- **Fragment is created through extending either Fragment class or the other existed Fragment sub-class**
- **To render an UI, onCreateView() method should be overridden.**
  - **If UI is not necessary, leave return null; within onCreateView() method**

```
public class ChatFragment extends Fragment {  
    @Override  
    protected View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
        View v = inflater.inflate(R.layout.chats, container, false);  
        return v;  
    }  
}
```



# Add Fragment into Activity

## ■ Two means

### □ Solution 1: Add Fragment within layout file

□ `<fragment></fragment>`

### □ Solutioni 2: Add Fragment when Activity running

□ Obtain one `FragmentManager` instance

□ Call `add()` method to add one Fragment

□ The first argument of `add()` is the `ViewGroup` where to input the Fragment (by `ResourceID`)

□ Then call `commit()` method to submit the act

```
DetailFragment details = new DetailFragment();  
FragmentManager ft = getSupportFragmentManager().beginTransaction();  
ft.add(android.R.id.content, details);  
ft.commit();
```

# Demo



```
<RelativeLayout 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="com.example.wechatinterfacechange.MainActivity">
    <fragment
        android:id="@+id/fragment"
        android:name="com.example.wechatinterfacechange.WeChat_Fragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_alignParentBottom="true"
        android:orientation="horizontal">
        <ImageView
            android:id="@+id/image1"
            android:layout_width="0dp"
            android:layout_height="50dp"
            android:layout_weight="1"
            android:scaleType="centerCrop"
            android:src="@drawable/bottom_1" />
        <ImageView
            android:id="@+id/image2"
            android:layout_width="0dp"
            android:layout_height="50dp"
            android:layout_weight="1"
            android:scaleType="centerCrop"
            android:src="@drawable/bottom_2" />
        <ImageView
            android:id="@+id/image3"
            android:layout_width="0dp"
            android:layout_height="50dp"
            android:layout_weight="1"
            android:scaleType="centerCrop"
            android:src="@drawable/bottom_3" />
        <ImageView
            android:id="@+id/image4"
            android:layout_width="0dp"
            android:layout_height="50dp"
            android:layout_weight="1"
            android:scaleType="centerCrop"
            android:src="@drawable/bottom_4" />
    </LinearLayout>
</RelativeLayout>
```



# Demo



```
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ImageView imageView1 = (ImageView)findViewById(R.id.image1);
        ImageView imageView2 = (ImageView)findViewById(R.id.image2);
        ImageView imageView3 = (ImageView)findViewById(R.id.image3);
        ImageView imageView4 = (ImageView)findViewById(R.id.image4);
        imageView1.setOnClickListener(l);
        imageView2.setOnClickListener(l);
        imageView3.setOnClickListener(l);
        imageView4.setOnClickListener(l);
    }
    View.OnClickListener l = new View.OnClickListener(){
        @Override
        public void onClick(View v) {
            FragmentManager fm = getFragmentManager();
            FragmentTransaction ft = fm.beginTransaction();
            Fragment f = null;
            switch(v.getId()){
                case R.id.image1:
                    f = new WeChat_Fragment();
                    break;
                case R.id.image2:
                    f = new Message_Fragment();
                    break;
                case R.id.image3:
                    f = new Find_Fragment();
                    break;
                case R.id.image4:
                    f = new Me_Fragment();
                    break;
                default:
                    break;
            }
            ft.replace(R.id.fragment, f);
            ft.commit();
        }
    };
}
```

# Demo



<RelativeLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent">
```

<ImageView

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:src="@drawable/weixin"
android:scaleType="fitXY"/>
```

</RelativeLayout>

```
public class Message_Fragment extends Fragment {
```

```
    @Nullable
```

```
    @Override
```

```
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
```

```
        View view = inflater.inflate(R.layout.message_fragment, null);
```

```
        return view;
```

```
    }
```

```
}
```

# Demo



```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/tongxun"
        android:scaleType="fitXY"/>
</RelativeLayout>
```

```
public class WeChat_Fragment extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
                             @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.wechat_fragment,null);
        return view;
    }
}
```

# Demo



```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/faxian"
        android:scaleType="fitXY"/>
</RelativeLayout>
```

```
public class Find_Fragment extends Fragment {
    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.find_fragment,null);
        return view;
    }
}
```

