

Button, Intent, Bundle

```

1  /*In MainActivity*/
2  btnreset.setOnClickListener(new View.OnClickListener() {
3      @Override
4      public void onClick(View v) {
5          Intent intent = new Intent(MainActivity.this,
6              SecondActivity.class);
7          Bundle bundle = new Bundle();
8          bundle.putString("name", name);
9          intent.putExtras(bundle);
10         startActivity(intent);
11         /*int requestCode = 2;
12         startActivityForResult(intent, requestCode);*/
13     }
14 });
15 @Override
16 protected void onActivityResult(int requestCode, int
17     resultCode, Intent data) {
18     String newNo = data.getStringExtra("_newNo");
19     switch (resultCode) {
20         case 1:
21             //
22     }
23 }
24 /*In SecondActivity*/
25 protected void onCreate(Bundle savedInstanceState) {
26     super.onCreate(savedInstanceState);
27     setContentView(R.layout.activity_second);
28     Bundle bundle = getIntent().getExtras();
29     String name = bundle.getString("name");
30 }
31 Intent intent = new Intent();
32 intent.putExtra("newNo", newNo);
33 int resultCode = 1;
34 SecondActivity.this.setResult(resultCode, intent);
35 SecondActivity.this.finish();

```

ListView, List, ArrayAdapter

```

1  private ListView fruitListView;
2  private List<Fruit> fruitList = new ArrayList<Fruit>();
3  fruitListView = (ListView) this.findViewById(R.id.
4      fruitListView);
5  fruitList.add(new Fruit("apple", R.mipmap.apple));
6  FruitAdapter adapter = new FruitAdapter(this,
7      R.layout.fruit_item, fruitList);
8  fruitListView.setAdapter(adapter);
9  fruitListView.setOnItemClickListener(new AdapterView.
10     OnItemClickListener() {
11         @Override
12         public void onItemClick(AdapterView<?> parent, View
13             view, int position, long id) {
14             ListView listView = (ListView) parent;
15             Fruit fruit = (Fruit) listView.getItemAtPosition
16                 (position);
17         }
18     });
19 fruitListView.setOnItemClickListener(new AdapterView
20     .OnItemLongClickListener() {
21         @Override
22         public boolean onItemLongClick(AdapterView<?> parent
23             , View view, int position, long id) {
24             try {
25                 ListView listView = (ListView) parent;
26                 Fruit fruit = (Fruit) listView.
27                     getItemAtPosition(position);
28                 FruitAdapter adapter = (FruitAdapter)
29                     listView.getAdapter();
30                 adapter.remove(fruit);
31                 adapter.notifyDataSetChanged();
32                 listView.invalidate();
33             } catch (Exception e) {e.printStackTrace();}
34             return false;
35         }
36     });
37 return true;
38 }
39 public class FruitAdapter extends ArrayAdapter<Fruit> {
40     private int resourceId;
41     public FruitAdapter(Context context, int
42         textViewResourceId, List<Fruit> fruits){
43         super(context, textViewResourceId, fruits);
44         this.resourceId = textViewResourceId;
45     }
46     @Override
47     public View getView(int position, View convertView,
48         ViewGroup parent){
49         Fruit fruit = getItem(position);
50         View view = LayoutInflater.from(this.getContext
51            ()).inflate(resourceId, null);
52         ImageView fruitImage = (ImageView) view.
53             findViewById(R.id.fruit_image);
54         TextView fruitName = (TextView) view.findViewById
55             (R.id.fruit_name);
56         fruitImage.setImageResource(fruit.getImageId());
57         fruitName.setText(fruit.getName());
58         return view;
59     }
60 }

```

AppWidgetProvider, BroadcastReceiver

```

1  public class MyWidgetProvider extends AppWidgetProvider
2  {
3      @Override
4      public void onUpdate(Context context,
5          AppWidgetManager appWidgetManager, int[]
6          appWidgetIds){
7          super.onUpdate(context, appWidgetManager,
8              appWidgetIds);
9          Intent intent = new Intent(context, MainActivity
10             .class);
11          PendingIntent pendingIntent = PendingIntent.
12              getActivity(context, 0, intent, 0);
13          RemoteViews remoteViews = new RemoteViews(
14              context.getPackageName(), R.layout.
15              widget_layout);

```

```

8          remoteViews.setOnClickPendingIntent(R.id.
9              widget_text, pendingIntent);
10         appWidgetManager.updateAppWidget(appWidgetIds,
11             remoteViews);
12     }
13 }
14 public class MyBroadcastReceiver extends
15     BroadcastReceiver {
16     @Override
17     public void onReceive(Context context, Intent intent
18     ){
19         RemoteViews remoteViews = new RemoteViews(
20             context.getPackageName(), R.layout.
21             widget_layout);
22         remoteViews.setTextViewText(R.id.widget_text,
23             intent.getStringExtra("message"));
24         AppWidgetManager.getInstance(context).
25             updateAppWidget(new ComponentName(context.
26                 getApplicationContext(), MyWidgetProvider.
27                 class), remoteViews);
28     }
29 }
30 private boolean isRegister;
31 private MyBroadcastReceiver myBroadcastReceiver;
32 protected void onCreate(Bundle savedInstanceState) {
33     myBroadcastReceiver = new MyBroadcastReceiver();
34     unregisterReceiver(myBroadcastReceiver);
35     registerReceiver(myBroadcastReceiver, new
36         IntentFilter(getResources().getString(R.string.
37             sysu)));
38     /* To Send a broadcast*/
39     Intent intent = new Intent(getResources().getString(
40         R.string.sysu));
41     intent.putExtra("message", mainEditText.getText().
42         toString());
43     sendBroadcast(intent);
44 }

```

Service

```

1  public class myService extends Service {
2      public final IBinder binder = new MyBinder();
3      public class MyBinder extends Binder{
4          myService getService(){
5              return myService.this;
6          }
7      }
8      @Override
9      public IBinder onBind(Intent intent) {
10         return binder;
11     }
12 }
13 private MusicService musicService = new musicService();
14 private ServiceConnection sc = new ServiceConnection() {
15     @Override
16     public void onServiceConnected(ComponentName
17         componentName, IBinder iBinder) {
18         musicService = ((MusicService.MyBinder)iBinder).
19             getService();
20     }
21     @Override
22     public void onServiceDisconnected(ComponentName
23         componentName) {
24         musicService = null;
25     }
26 };
27 private void bindServiceConnection() {
28     Intent intent = new Intent(MainActivity.this,
29         MusicService.class);
30     startService(intent);
31     bindService(intent, sc, this.BIND_AUTO_CREATE);
32 }

```

SharedPreferences, CheckBox, AutoCompleteTextView

```

1  SharedPreferences sp = getSharedPreferences("xmlUserData
2      ", MODE_PRIVATE);
3  SharedPreferences.Editor editor = sp.edit();
4  editor.putString("username", usr);
5  editor.commit();
6  checkboxRem.setOnCheckedChangeListener(new CompoundButton.
7      OnCheckedChangeListener() {
8          @Override
9          public void onCheckedChanged(CompoundButton
10             compoundButton, boolean isChecked) {
11             editor.putBoolean("isChecked", isChecked);
12             editor.commit();
13             if(isChecked && edtUsr.getText().toString().
14                 equals(sp.getString("username", null))) {
15                 edtPwd.setText(sp.getString("password", null
16                     ));
17             }
18         }
19     });
20 checkboxRem.setChecked(true);
21 AutoCompleteTextView fileName = (AutoCompleteTextView)
22     this.findViewById(R.id.textviewFileName);
23 fileName.setAdapter(new ArrayAdapter<String>(this,
24     android.R.layout.simple_dropdown_item_1line, this.
25     fileList()));

```

SQLite

```

1  public class MyDatabaseHelper extends SQLiteOpenHelper {
2      private static final String DB_NAME = "Contacts.db";
3      private static final String TABLE_NAME = "Contacts";
4      private static final int DB_VERSION = 1;
5      public MyDatabaseHelper(Context context) {
6          super(context, DB_NAME, null, DB_VERSION);
7      }
8      @Override
9      public void onCreate(SQLiteDatabase db) {
10         String CREATE_TABLE = "create table " +
11             TABLE_NAME

```

```

11         + " (_id integer primary key
12             autoincrement, "
13         + "no text not null, "
14         + "name text not null, "
15         + "pnumber text);";
16     db.execSQL(CREATE_TABLE);
17 }
18 @Override
19 public void onUpgrade(SQLiteDatabase db, int
    oldVersion, int newVersion) {
20     String DROP_TABLE = "DROP TABLE IF EXISTS " +
21     TABLE_NAME;
22     db.execSQL(DROP_TABLE);
23     onCreate(db);
24 }
25 public long insert(Contact entity) {
26     SQLiteDatabase db = getWritableDatabase();
27     ContentValues values = new ContentValues();
28     values.put("_no", entity.getNo());
29     values.put("_name", entity.getName());
30     values.put("_pnumber", entity.getPhoneNumber());
31     long id = db.insert(TABLE_NAME, null, values);
32     db.close();
33     return id;
34 }
35 public int update(Contact entity) {
36     SQLiteDatabase db = getWritableDatabase();
37     String whereClause = "no = ?";
38     String[] whereArgs = {entity.getNo()};
39     ContentValues values = new ContentValues();
40     values.put("_no", entity.getNo());
41     values.put("_name", entity.getName());
42     values.put("_pnumber", entity.getPhoneNumber());
43     int rows = db.update(TABLE_NAME, values,
44     whereClause, whereArgs);
45     db.close();
46     return rows;
47 }
48 public int delete(Contact entity) {
49     SQLiteDatabase db = getWritableDatabase();
50     String whereClause = "no = ?";
51     String[] whereArgs = {entity.getNo()};
52     int rows = db.delete(TABLE_NAME, whereClause,
53     whereArgs);
54     db.close();
55     return rows;
56 }
57 public Cursor query() {
58     SQLiteDatabase db = getReadableDatabase();
59     return db.query(TABLE_NAME, null, null, null,
60     null, null, null);
61 }
62 }
63 public class MySimpleAdapter extends SimpleAdapter {
64     private ArrayList<Map<String, String>> mData;
65     public MySimpleAdapter(Context context, List<?
66     extends Map<String, ?>> data, int resource,
67     String[] from, int[] to) {
68         super(context, data, resource, from, to);
69         this.mData = (ArrayList<Map<String, String>>)
70         data;
71     }
72     @Override
73     public View getView(int position, View convertView,
74     ViewGroup parent) {
75         final int mPosition = position;
76         return super.getView(position, convertView,
77         parent);
78     }
79 }
80 /*In MainActivity*/
81 private MyDatabaseHelper myDatabaseHelper = new
82     MyDatabaseHelper(MainActivity.this);
83 private List<Map<String, String>> dataList = new
84     ArrayList<Map<String, String>>();
85 listView.setOnItemClickListener(new AdapterView.
86     OnItemClickListener() {
87     @Override
88     public boolean onItemClick(AdapterView<?> parent,
89     View view, int position, long id) {
90         ListView listView = (ListView) parent;
91         final HashMap<String, String> map = (HashMap<String,
92         String>)listView.getItemAtPosition(position);
93         map.get("no");
94         AlertDialog...
95         return true;
96     }
97 });
98 private void setData(List<Map<String, String>> mDataList
99 ) {
100     Map<String, String> mData;
101     Cursor c = myDatabaseHelper.query();
102     while (c.moveToNext()) {
103         mData = new HashMap<String, String>();
104         mData.put("no", c.getString(c.getColumnIndex("
105         no")));...
106         mDataList.add(mData);
107     }
108 }
109 private void updateListView() {
110     dataList.clear();
111     setData(dataList);
112     MySimpleAdapter mySimpleAdapter = new
113     MySimpleAdapter(this, dataList, R.layout.
114     contact_item,
115     new String[] { "no", "name", "pnumber" },
116     new int[] { R.id.contact_no, R.id.
117     contact_name, R.id.contact_phonenumber
118     });
119     ListView lv = (ListView) this.findViewById(R.id.
120     lv_contact);
121     lv.setAdapter(mySimpleAdapter);
122 }

```

— Message, Handler, Thread, Runnable —

```

2 private static final int TIME_OUT = 1;
3 private Handler handler = new Handler() {
4     @Override
5     public void handleMessage(Message message) {
6         switch (message.what) {
7             case UPDATE_CONTENT: ...
8             default: super.handleMessage(message);
9         }
10     }
11 };
12 private void aFunction() {
13     Thread workThread = new Thread(new Runnable() {
14     @Override
15     public void run() {
16         Message message = new Message();
17         message.what = UPDATE_CONTENT;
18         message.obj = parseXMLWithPull(response.toString
19         ());
20         handler.sendMessage(message);
21     }
22     workThread.start();
23 });

```

— Utils —

```

1 public android.os.Handler handler = new android.os.
2     Handler();
3 public Runnable runnable = new Runnable() {
4     @Override
5     public void run() {
6         handler.postDelayed(runnable, 100);
7     }
8 };
9 handler.removeCallbacks(runnable);
10 InputMethodManager imm = (InputMethodManager)
11     getSystemService(Context.INPUT_METHOD_SERVICE);
12 if(imm.isActive()) {
13     imm.hideSoftInputFromWindow(EditFileActivity.this.
14     getCurrentFocus().getWindowToken(),
15     InputMethodManager.HIDE_NOT_ALWAYS);
16 }
17 @Override
18 public boolean onKeyDown(int keyCode, KeyEvent event) {
19     if(keyCode == KeyEvent.KEYCODE_BACK) {...}
20     return true;
21 }
22 AlertDialog.Builder builder = new AlertDialog.Builder(
23     MainActivity.this);
24 builder.setMessage("Delete?");
25 builder.setTitle("Hint");
26 builder.setPositiveButton("Confirm", new DialogInterface
27     .OnClickListener() {
28     @Override
29     public void onClick(DialogInterface dialogInterface,
30     int i) {...}
31 });
32 builder.setNegativeButton("Cancel", new DialogInterface.
33     OnClickListener() {
34     @Override
35     public void onClick(DialogInterface dialogInterface,
36     int i) {...}
37 });
38 builder.show();
39 String newNo = "32654984";
40 newNo.matches("^([1-9]\\d{7})$");

```

— AndroidManifest.xml —

```

1 <manifest xmlns:android="http://schemas.android.com/apk/
2     res/android" package="com.wsine.west.finalexam" >
3     <uses-permission android:name="android.permission.
4     WRITE_EXTERNAL_STORAGE"/>
5     <uses-permission android:name="android.permission.
6     READ_EXTERNAL_STORAGE"/>
7     <uses-permission android:name="android.permission.
8     ACCESS_NETWORK_STATE"/>
9     <uses-permission android:name="android.permission.
10     INTERNET"/>
11     <uses-permission android:name="com.android.launcher.
12     permission.READ_SETTINGS"/>
13     <uses-permission android:name="android.permission.
14     WAKE_LOCK"/>
15     <uses-permission android:name="android.permission.
16     WRITE_SETTINGS"/>
17 <application
18     android:allowBackup="true" >
19     <activity
20         android:name=".MainActivity"
21         android:label="@string/app_name" >
22         <intent-filter>
23             <action android:name="android.intent.action.
24             MAIN"/>
25             <category android:name="android.intent.
26             category.LAUNCHER"/>
27         </intent-filter>
28     </activity>
29     <activity
30         android:name=".SecondActivity"
31         android:label="@string/title_activity_detail"
32         android:launchMode="singleInstance" >
33     </activity>
34     <service android:name=".MyService" android:exported=
35     "true"></service>
36     <receiver
37         android:name=".MyWidgetProvider"
38         android:label="@string/app_name" >
39         <intent-filter>
40             <action android:name="android.appwidget.
41             action.APPWIDGET_UPDATE"/>
42         </intent-filter>
43         <meta-data
44             android:name="android.appwidget.provider"
45             android:resource="@xml/widget_provider"/>
46     </receiver>
47 </application>
48 </manifest>

```