Development with Android

Computing Science and Mathematics
University of Stirling

Intents and Intent Filters

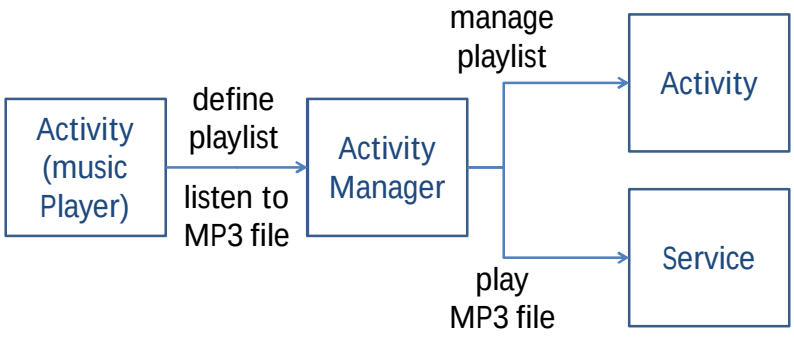
Intents

- An *Intent* describes what is to be done, e.g. *VIEW*, *CALL* or *PLAY*
- Android matches an *Intent* with an *Activity* that can best perform the work
- An *Activity* or a *BroadcastReceiver* describe what *Intents* they can service in their *IntentFilters* (defined in *AndroidManifest.xml*)
- Intents can start Activities which are specified *explicitly* (name a class) or *implicitly* (request action to be performed on some data).

Intents

- Intents work similar to “events”
- Intents encourages decoupling of components and allows for late binding and extending an application’s functionality
- Android broadcasts system events (incoming phone call, SMS message received, battery low)
  - Native apps simply listen for these events and process them accordingly
  - Can deploy custom apps listening for these events (intents) and replace native apps

Using Intents



5

Intent Context

Context Methods	Action
startActivity(Intent) *startActivityForResult(Intent, int)*	Launch an *Activity* (for some request code)
sendBroadcast(Intent)	Send intent to a relevant *BroadcastReceiver*
startService(Intent) *bindService(Intent, ServiceConnection, int)*	Start a *Service* (or bind to an existing one for some request code)

6

Intent Data

Primary Attributes (Mandatory)	Secondary Attributes (Optional)
Action: to be taken	*Category:* additional information
Data: for operation (in URI form)	*Type:* MIME type of data
	Component: specifies component explicitly (an explicit intent)
	Extras: additional information in *Bundle* form

Intent Example – Raising Intent

```
public class MyActivity extends Activity {

 static final int PICK_CONTACT_REQUEST = 1;

 // select contact on pressing centre pad button
 public boolean onKeyDown(int keyCode, KeyEvent event) {
  // listen for directional pad centre button
  if (keyCode == KeyEvent.KEYCODE_DPAD_CENTER) {
   // let user pick a contact, starting subactivity
       Intent pickContactIntent = new Intent(Intent.ACTION_PICK,
                                                 Uri.parse("content://contacts"));
       pickContactIntent.setType(Phone.CONTENT_TYPE);
                    // Show user only contacts w/ phone numbers
       startActivityForResult(pickContactIntent, PICK_CONTACT_REQUEST);
   return(true);
  }
  return(false);
 }
```

ACTION_PICK: launches an activity that lets you pick an item from the Content Provider specified by the data URI, result will be returned.

Intent Filter – Handling an Intent

- define an Intent Filter in the manifest file
- application components can declare the actions they support

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="interlife.stir.ac.uk"
<application android:icon="@drawable/icon" android:label="@string/app_name">
      <activity android:name="erick.stir.ac.uk.ContactSelector"
            android:label="@string/app_name">
        <intent-filter>
          <action android:name="android.intent.action.PICK" />
          <category android:name="android.intent.category.DEFAULT" />
          <data android:path="contacts" android:scheme="content" />
        </intent-filter>
      </activity>
  </application>
</manifest>
```


Intent – Returning a value

```
public class MyOtherActivity extends Activity {

Button okButton = (Button) findViewById(R.id.ok_button);
okButton.setOnClickListener(new View.OnClickListener() {
 public void onClick(View view) {
  long selected_entry_id = listView.getSelectedItemId();
  Uri selectedEntry = Uri.parse("content://contacts/" + selected_entry_id);
  Intent result = new Intent(Intent.ACTION_PICK, selectedEntry);
  setResult(RESULT_OK, result);
  finish();
}});

Button cancelButton = (Button) findViewById(R.id.cancel_button);
cancelButton.setOnClickListener(new View.OnClickListener() {
 public void onClick(View view) {
 setResult(RESULT_CANCELED);
  finish();
}});
```

Return a value as an Intent, e.g. intent contains the selected value from a list
setResult takes the "result code" – RESULT_OK and RESULT_CANCELED are predefined, can be any integer number.

Intent – Receiving a returned value

```
// handle result of contact request
protected void onActivityResult(int requestCode, int resultCode,
 Intent data) {
  if (requestCode == PICK_CONTACT_REQUEST) {
    if (resultCode == RESULT_OK)
      // a contact was picked – display it to the user
      startActivity(new Intent(Intent.ACTION_VIEW, data));
  }
 }

}
```

ACTION_VIEW: one of the most common intents, asks to display the data provided in the most reasonable manner (depending on the URI schema of the data).
http: - web browser
tel: - dialer
contact: - contact manager

Action-Data Pairs

- *ACTION_VIEW content://contacts/people/1*
  - display information about person with identifier *1*
- *ACTION_DIAL content://contacts/people/1*
  - display phone dialler with person *1* filled in
- *ACTION_VIEW tel:1234567*
  - display phone dialler with *1234567* filled in
- *ACTION_DIAL tel:1234567*
  - display phone dialler with the number filled in.
- *ACTION_CALL tel:1234567*
  - calls number directly (needs CALL_PHONE permission set in the manifest file)

Intents and Chained Activities

- *ACTION_EDIT content://contacts/people/1* :
  - edit information about person with identifier *1*
- *ACTION_VIEW content://contacts/people* :
  - display all contacts (with the *Contacts* app)
  - selecting contact *N* results in a new intent *ACTION_VIEW content://contacts/people/N*

Dialling using An Intent

- Start activity with a single button press:
  - phone activity is launched
  - given number appears
- Implicit or explicit Intent can be used

Implicit Intent - 1

```
public class ImplicitIntent extends Activity implements
 OnClickListener {

  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    Button dial = (Button) findViewById(R.id.dial);
    dial.setOnClickListener(this);
  }
```


Implicit Intent - 2

```
  // start activity with request code and data
  public void onClick(View view) {
    Intent call = new Intent(Intent.ACTION_DIAL,
      Uri.parse("tel:1234567"));
    startActivity(call);
  }

  // or set action and data separately
  public void onClick(View view) {
    Intent call = new Intent();
    call.setAction(Intent.ACTION_DIAL);
    call.setData(Uri.parse("tel:1234567"));
    startActivity(call);
  }
```

Use Intent.ACTION_DIAL instead of Intent.ACTION_CALL. This shows the dialer with the number already entered, but allows the user to decide whether to actually make the call or not.
ACTION_DIAL does not require the CALL_PHONE permission in the manifest file.

Using An Explicit Intent

- Explicit intents are used to invoke specific components
- In this example, clicking ‘Go’ will launch a ‘Hello’ app

Explicit Intent

```
public class ExplicitIntent extends Activity implements
 View.OnClickListener {

public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.main);
   Button go = (Button) findViewById(R.id.go);
   go.setOnClickListener(this);
 }

public void onClick(View view) {
   Intent go = new Intent();
  //specify an application and activity explicitly
   go.setClassName("uk.ac.stir.cs.android",
     "uk.ac.stir.cs.android.Hello");
   startActivity(go);
 }
```

Resolve Intent

- Using 3rd party components is unreliable
  - are the expected/required apps actually available on the phone?
- Check if your Intent will resolve to an Activity before calling startActivity()
- PackageManager can be queried to find Activities which will be launched based on an Intent

```
Intent intent = new Intent(Intent.ACTION_PICK, new Uri("content://contacts"));
PackageManager pm = getPackageManager();
ComponentName cn = intent.resolveActivity(pm);
if(cn == null) { //no suitable Activity found, check if Google Play is available
    Uri marketUri = Uri.parse("market://search?pname:com.myapp.packagename");
     Intent marketIntent = new Intent(Intent.ACTION_VIEW).setData(marketUri);
     if(marketIntent.resolveActivity(pm) != null)   //Google Play available?
          startActivity(marketIntent);      //Yes! Start Google Play to find a suitable app
      else Log.d(TAG, "Google Play not available.");   //No Google Play
 } else startActivity(intent); //suitable Activity found, start it
}
```