



Intents, Intent Filters, Broadcast Receivers

Miroslav Mironov





Previously on *Android OS Programming*

- Views.
- Activities.
- Services.





Intents





What are intents?

- A message-passing mechanism.
- Used to start application components.
- Used to broadcast messages across the system.



Using intents to launch activities

Explicitly starting a new activity

```
Intent i = new
    Intent (MainActivity.this,
        SignUpActivity.class);

startActivity(i);
```



Using intents to launch activities

Implicitly starting a new activity

```
Intent i = new
    Intent("com.fmi.androidcourse
        .SIGN_UP");

startActivity(i);
```





Launching Sub-Activities

A ***request code***, necessary for a later identification of the sub-Activity that has returned a result.

```
private static final int CODE = 1;
Uri uri =
    Uri.parse("content://contacts/people");
Intent i = new Intent(Intent.ACTION_PICK, uri);
startActivityForResult(i, CODE);
```

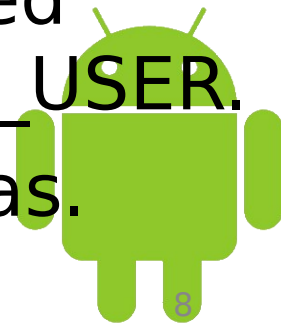




Returning results

```
Intent result = new Intent(null);
result.putExtra("userSelected", user);
setResult(RESULT_OK, result);
finish();
```

- **result code** – Activity.RESULT_OK, Activity.RESULT_CANCELED, a user-defined integer starting at Activity.RESULT_FIRST_USER.
- **result** – an intent to store data and extras.



Handling Sub-Activity Results

- When a sub-Activity closes, its parent Activity's `onActivityResult` event handler is fired.
- ```
public void onActivityResult (
 int requestCode,
 int resultCode,
 Intent data);
```



## Handling Sub-Activity Results

```

switch (requestCode) {
 case SHOW_ACTIVITY_ONE :
 if (resultCode == RESULT_OK) {
 Bundle b = data.getExtras();
 String un = b.getStringExtra("username");
 }
 break;
 case SHOW_ACTIVITY_TWO :
 ...

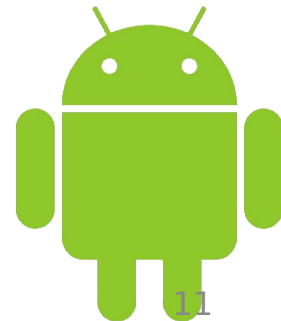
```





## Native Android Actions

- **ACTION\_ANSWER**
- **ACTION\_DELETE**
- **ACTION\_DIAL**
- **ACTION\_PICK**
- **ACTION\_VIEW**
- **ACTION\_WEB\_SEARCH**





## Intent Filters





## Intent Filters

- Used to register Activities, Services and Broadcast Receivers as being capable of performing an action on a particular kind of data.
- To register a component, use the `intent-filter` tag in the component's manifest node.



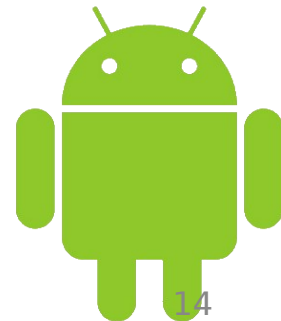


## The `<intent-filter>` tag

```
<action
```

```
 android:name="com.fmi.androidcourse.SIGN_UP"
/>
```

- **android:name** - a unique string, identifying the action being served. Best practice is to use the package naming conventions in Java.





## The `<intent-filter>` tag

```
<category
 android:name="android.intent.category.DEFAULT"
/>
```

- **android:name** – specifies under which circumstances the action should be serviced. Some categories are:  
BROWSABLE, DEFAULT, GADGET,  
HOME, LAUNCHER





## The `<intent-filter>` tag

- The **data** tag - specifies matches for data a component can act on.

`<data attribute="..." />` where *attribute* might be one of the following: `android:host`, `android:path`, `android:port`, `android:scheme`, `android:mimetype`







```
onNewIntent (Intent intent) ;
```

- Called for activities with launch mode set to “singleTop” when the activity is started again.
- *intent* - the intent that re-launched the activity.
- Use `setIntent (intent)` to update the original intent that started the activity.





## Now, a demo...

The demo can be found on the course Moodle web page.





## Intent Resolution

- The best intent filter match possible is found by the following process:
  1. *Android makes a list of all intent filters available in the installed packages.*
  2. *The intent filters that do not match the action and the category of the intent being resolved, are removed from the list.*





## Intent Resolution

3. *Each part of the intent's data URI is compared to the intent filter's data tag. Any mismatches will remove the intent filter from the list.*
4. *If more than one component is resolved, then they are ordered by priority and the component with highest priority is returned.*





## Passing on Responsibility

- Pass responsibility for action handling to the next best matching application component.

```
Intent i = getIntent();
if (condition) {
 startNextMatchingActivity(i);
}
```





## Broadcast Receivers





## Broadcast Intents

- Notify listeners for system or application events.
- Help make your application more open.
- Can be listened for to react to system changes and application events.





## Broadcasting Events

1. Construct the intent you want to broadcast (set the action, data, category and extras).
2. Send it using the **sendBroadcast** method.







## Broadcast Receivers

- Listen for broadcast intents.
- Need to be registered either in code or within the application manifest.
- Extend the **BroadcastReceiver** class.
- Override the **onReceive** event.





## The `onReceive` event

- Executes when a broadcast intent matching the intent filter for the receiver turns up.
- Must complete within **10** seconds. Sometimes this limit mistakenly thought to be 5 seconds.



## Native Android Broadcast Actions

- **ACTION\_CAMERA\_BUTTON**
- **ACTION\_DATE\_CHANGED**
- **ACTION\_TIME\_CHANGED**
- **ACTION\_GTALK\_SERVICE\_CONNECTED**
- **ACTION\_MEDIA\_MOUNTED (UNMOUNTED)**
- **ACTION\_SCREEN\_OFF (ON)**





## Broadcast receivers in practice...

The demo can be found on the course Moodle  
web page.





Time for your questions  
???

