



# Intents and Broadcast Receivers

*Dimitar G. Dimitrov<sup>1,2</sup>*

*<sup>1</sup>Astea Solutions AD*

*<sup>2</sup>Sofia University*





## Motivation

- Mobile apps are sandboxed
- Strict limits on their interaction with the hardware and native components
- Android apps can use Intents, Broadcast Receivers, Adapters, Content Providers, and the Internet to extend beyond those boundaries



אנדרואיד



# Intents





## What are Intents?

- A message-passing mechanism
- An intent is an abstract description of an operation to be performed
- Used to start application components (activities, services)
- Used to broadcast messages across the system



## Example: using intents to launch activities

- **Explicitly** starting a new activity

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

startActivity(i);
```



## Example: using intents to launch activities

- **Implicitly** starting a new activity

```
Intent i = new Intent(
    "net.astezasolutions.aos.SIGN_UP");

startActivity(i);
```





## Launching Sub-Activities

- A **request code** is 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; //do you tend to forget it? :)
case SHOW_ACTIVITY_TWO :
    ...

```





## Intent Structure

- The primary pieces of information in an intent are:
  - **action** - The general action to be performed, such as ACTION\_VIEW, ACTION\_EDIT
  - **data** - The data to operate on, such as a person record in the contacts database, expressed as a Uri
    - URI structure: scheme://host:port/path





# Native Android Actions and Corresponding Data

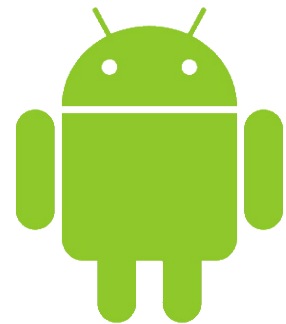
- **`ACTION_VIEW`** *content://contacts/people/1*
  - Display information about the person whose id is "1"
- **`ACTION_DIAL`** *content://contacts/people/1*
  - Display the phone dialer with the person filled in
- **`ACTION_VIEW`** *tel:123*
  - Display the phone dialer with the given number filled in
  - Note how the VIEW action does what what is considered the most reasonable thing for a particular URI
- **`ACTION_DIAL`** *tel:123*
  - Display the phone dialer with the given number filled in
- And many more



אנדרואיד



# 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="net.asteasolutions.aos.SIGN_UP"
/>
```

- `android:name` - a unique string, identifying the action being served
- An intent filter must contain `action` tag
- Best practice is to use the package naming conventions in Java





## The `<intent-filter>` tag

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

- Categories give additional information about the action to execute
- Some categories are: BROWSABLE, DEFAULT, GADGET, HOME, LAUNCHER







## The `<intent-filter>` tag

- The **data** tag specifies matches for data a component can act on
- There are separate optional attributes for each part of the URI, as well as for MIME type

```
<data android:scheme="..."
      android:host="..."
      android:port="..."
      android:path="..."
      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





# 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 (cont.)

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

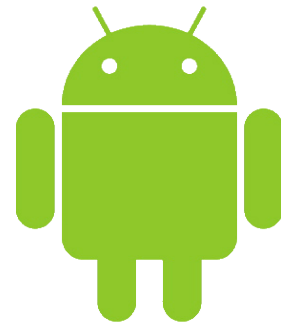




## Passing on Responsibility

- To pass responsibility for action handling to the next best matching application component:

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



אנדרואיד



# 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

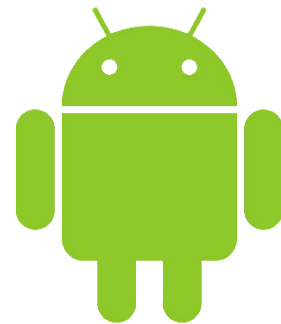
- Listen for broadcast intents
- Need to be registered either in code or within the application manifest
- Extend the `BroadcastReceiver` class
- Override the `onReceive` method





## 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 (TIME, TIMEZONE)`
- `ACTION_MEDIA_MOUNTED (UNMOUNTED)`
- `ACTION_SCREEN_OFF (ON)`
- And many more





## Q&A + Feedback

- Questions?
- Feedback section:
  - Did you hear well?
  - Was there anything you didn't understand?
  - What would you like changed in our next lecture?

