Intents


Repo: https://github.com/karlmorris/AndroidIntents
Overview

- Android's intent system
- Intent makeup
- Common intents
- Intent filters and resolution
Intents

• A messaging object used to request an action
• Generally used to
  – Start an activity
  – Start a service
  – Send a broadcast
• Two types of intents
  – Explicit - specify a component
  – Implicit - do not specify a component
Intents in Action

An intent passed between two activities
Anatomy of an Intent

- Component name
- Action
- Data
- Category
- Extras
- Flags
Component

- Optional
  - Intent is explicit if present, implicit if not
- e.g.
  - Activity
  - Service
Action

- A string denoting a generic action to perform
- For broadcasts, it is usually what took place
- Predefined actions available or define your own
- Usually determines how the rest of the intent is built
- e.g.
  - ACTION_SEND
    - Use this action when you have some data that can be shared through another app
Data

• A pointer to the data to be acted on and/or the data's MIME type

• MIME type useful to assist Android in identifying best component to handle data
Category

- A string providing more information on the component that should handle the intent
- Not required by most intents
- e.g.
  - CATEGORY_BROWSABLE
    - The target activity allows itself to be started by a web browser to display data referenced by a link
  - CATEGORY_LAUNCHER
    - The activity is the initial activity of a task and is listed in the system's application launcher
Extras

• Key-value pairs required to complete requested action
• Many EXTRA_* keys defined by framework
• Does not assist in helping resolve handling component
• e.g.
  – EXTRA_EMAIL and EXTRA_SUBJECT - provide email addresses and subject respectively if using ACTION_SEND to send an email
Flags

• Metadata for intent

• Can provide Android with launch and post-launch instructions for activities

• e.g.
  – Launch modes
    • FLAG_ACTIVITY_SINGLE_TOP
  – Post-launch instructions
    • FLAG_ACTIVITY_TASK_ON_HOME
    • FLAG_ACTIVITY_NO_HISTORY
Example of an Explicit Intent

• Open another activity

// Start an activity in the same application
Intent sendIntent = new Intent(this, Component.class);
sendIntent.putExtra(keyString, valueString);

startActivity(sendIntent);
Example of an Implicit Intent

- Create and request sending a text message

```java
// Create the text message with a string
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, textMessage);
sendIntent.setType("text/plain");

// Verify that the intent will resolve to an activity
if (sendIntent.resolveActivity(getPackageManager()) != null) {
    startActivity(sendIntent);
}
```
Responses to Implicit Intents

- Multiple components may be able to handle an intent
- Android presents a chooser
  - User may set default
- A chooser can be programmatically launched
Examples of Common Actions

- **Time and Date**
  - ACTION_SET_ALARM, ACTION_SET_TIMER, ACTION_SHOW_ALARMS, ACTION_INSERT

- **Media**
  - ACTION_IMAGE_CAPTURE, ACTION_VIDEO_CAPTURE

- **Email**
  - ACTION_SENDTO, ACTION_SEND, ACTION_SEND_MULTIPLE

- **Files**
  - ACTION_GET_CONTENT, ACTION_OPENDOCUMENT, ACTION_CREATEDOCUMENT

- **Contacts and People**
  - ACTION_PICK, ACTION_INSERT
Examples of Common Actions

- Web browsing
  - ACTION_VIEW
- Show a location on a map
  - ACTION_VIEW
- Play a media file
  - ACTION_VIEW
- See details of a specific contact
  - ACTION_VIEW
Intent Filters

• An intent filter is an declaration in an app's manifest that specifies the types of implicit intents the component would like to receive

• `<intent-filter>` elements are added as children of an `<activity>` element

• An activity can declare zero or more intent filters. If no intent filters are listed, that activity can only respond to explicit intents
Intent Resolution

- Android attempts to find the best activity to handle implicit intents
- This is done by comparing the intent to the declared intent filters of installed applications
- Three aspects are considered in order
  - Intent action
  - Intent data
  - Intent category
Action Test

- An intent filter can declare zero or more `<action>` elements
  
  ```xml
  <intent-filter>
    <action android:name="android.intent.action.EDIT" />
    <action android:name="android.intent.action.VIEW" />
    ...
  </intent-filter>
  
  - An intent must match one of the actions to get through the filter
  
  - If a filter does not list any actions then all intents fail, unless an intent itself does not specify an action
Data Test

• An intent filter can declare zero or more <data> elements

```xml
<intent-filter>
  <data android:mimeType="text/html" android:scheme="http" ... />
  <data android:mimeType="text/html" android:scheme="https" ... />
  ...
</intent-filter>
```

• There are various components of a URI that can be matched
  – scheme, host, port, and path

• If a filter lists a mime type, both mime type and URI components must match for the intent to pass the filter
Category Test

• An intent filter can declare zero or more <category> elements

  <intent-filter>
   <category android:name="android.intent.category.DEFAULT" />
   <category android:name="android.intent.category.BROWSABLE" />
   ...
  </intent-filter>

• A filter must match all the categories in an intent for it to pass
  – The reverse is not true

• Android automatically adds CATEGORY_DEFAULT to all implicit intents
  – Your Activity's filter must include android.intent.category.DEFAULT to receive implicit intents
Example Intent Filter

<activity android:name=".BrowserActivity">

  <intent-filter>

    <action android:name="android.intent.action.VIEW" />

    <!-- Include the host attribute if you want your app to respond only to URLs for a specific domain. -->
    <data android:scheme="http" android:host="www.mydomain.com" />

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

    <!-- The BROWSABLE category is required to get links from webpages. -->
    <category android:name="android.intent.category.BROWSABLE" />

  </intent-filter>

</activity>