

User Interaction: Intro to Android

Assoc. Professor Donald J. Patterson
INF 134 Fall 2012



Intents

- Activities, Services and Broadcast Receivers are activated through messages, called **Intents**.
- What do I mean by “messages”?
 - Late-run time binding between components in the same or different applications.
 - An abstract method call
 - Sent between apps on the phone
- An **Intent** is actually an object holding data about the message

<http://developer.android.com/guide/components/intents-filters.html>

Intents

- Examples:

- If you want to send an Intent you might do the following:

- `startActivity(new Intent(Intent.ACTION_DIAL,"tel:5555555555"));`

- If you want to be notified when the user has disconnected the power cord you might create a BroadcastReceiver that registers for the "ACTION_POWER_DISCONNECTED" Intent.

- (Look at all of the Intents)

<http://developer.android.com/reference/android/content/Intent.html>

Intents

- An **Intent** object is passed to `Context.startActivity()` or `Activity.startActivityForResult()` to launch an activity or get an existing activity to do something new.
- An **Intent** object is passed to `Context.startService()` to initiate a service or deliver new instructions to an ongoing service.
- **Intent** objects passed to any of the broadcast methods (such as `Context.sendBroadcast()`) are delivered to all interested broadcast receivers. Many kinds of broadcasts originate in system code.

<http://developer.android.com/guide/components/intents-filters.html>

Intents

- In each case, the Android system finds the appropriate activity, service, or set of broadcast receivers to respond to the **intent**, instantiating them if necessary. There is no overlap within these messaging systems: Broadcast intents are delivered only to broadcast receivers, never to activities or services. An intent passed to **startActivity()** is delivered only to an activity, never to a service or broadcast receiver, and so on.

<http://developer.android.com/guide/components/intents-filters.html>

Intents

- Android uses rules to figure out which components should handle an **intent**. For intents that don't explicitly name a target component, this process involves testing the Intent object against intent filters associated with potential targets.
- We are going to create a service which is going to handle all InputMethod Intents
- We will catch them with an intent-filter
- You will get skeleton code and implement a text input method

<http://developer.android.com/guide/components/intents-filters.html>

Intent Resolution

- **Explicit** Intents
 - This names a target component by name
 - This is usually used within an application because generic components' specific names aren't know in advance
- **Implicit** Intents
 - Do not name a specific component
 - They are asking for types of services

<http://developer.android.com/guide/components/intents-filters.html>

Intent Resolution

- **Implicit** Intents
 - Requires Android to use a strategy to pick the best component
 - Android looks at the **Intent Filters** of a component to figure out if it should pass the intent or not
 - **Filters** advertise what the component is designed to handle
 - They open the component to receiving **Intents** from Android
 - Without a filter a component can only get **Explicit Intents**
 - **Explicit Intents** do not interact with the **Filter**

<http://developer.android.com/guide/components/intents-filters.html>

Intent Resolution

- **Implicit** Intents
 - **Intent Filters** are usually set up in the Android manifest xml file
 - A component can have several **filters**
 - An **Intent** must pass through three components of a **filter** for Android to activate the component
 - Action
 - Data
 - Category

<http://developer.android.com/guide/components/intents-filters.html>

Intent Resolution

- **Action**

```
<intent-filter . . . >
  <action android:name="com.example.project.SHOW_CURRENT" />
  <action android:name="com.example.project.SHOW_RECENT" />
  <action android:name="com.example.project.SHOW_PENDING" />
  . . .
</intent-filter>
```

- An Intent must match at least 1 for the filter to pass
 - (OR)
- If there are no actions the filter only matches an Intent with no action

<http://developer.android.com/guide/components/intents-filters.html>

Intent Resolution

- **Category**

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

- An Intent must match all categories for the filter to pass
 - (AND)
- An Intent-filter with no category matches every Intent category test (with some nuances)

<http://developer.android.com/guide/components/intents-filters.html>

Intent Resolution

- **Data**

```
<intent-filter . . . >
  <data android:mimeType="video/mpeg" android:scheme="http" . . . />
  <data android:mimeType="audio/mpeg" android:scheme="http" . . . />
  . . .
</intent-filter>
```

- **mimeType, scheme, host, port, path**
- An Intent must match all declared categories in 1 tag for the filter to pass
 - (OR on tags, AND on attributes)

<http://developer.android.com/guide/components/intents-filters.html>

Intent Resolution

- **Examples**

- Respond to all intents that operate on images:

```
<data android:mimeType="image/*" />
```

- Respond to all intents that operate on internet video

```
<data android:scheme="http" android:type="video/*" />
```

<http://developer.android.com/guide/components/intents-filters.html>

Intent Resolution

- Filters are also help Android manage the user experience

```
<action android:name="android.view.InputMethod" />
```

- For Example,
 - Letting the user know that another keyboard is available

<http://developer.android.com/guide/components/intents-filters.html>

Things to know for Assignment 02

- LogCat
 - This enables you to output console messages
 - Only when in debug mode!

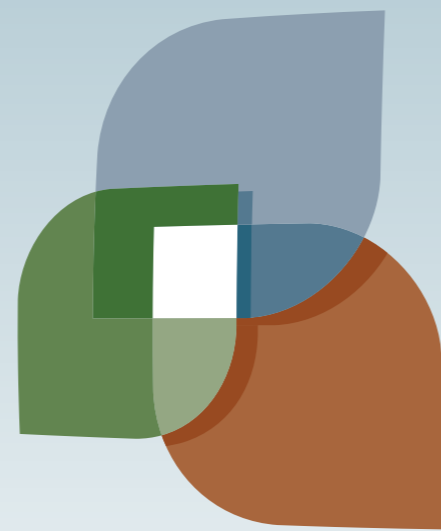
<http://developer.android.com/guide/components/intents-filters.html>

Things to know for Assignment 02

- Turn off auto-rotate
 - In preferences, not in code



<http://developer.android.com/guide/components/intents-filters.html>



L U C I

