Android Developer Fundamentals

Activities and Intents

Lesson 2

2.3 Starting Activities with Implicit Intents

Contents

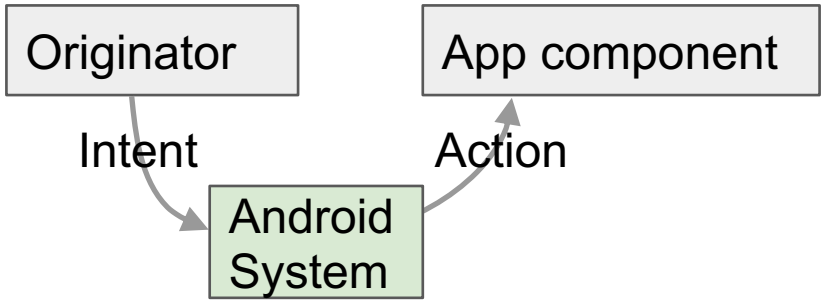
- Intents—recap
- Implicit intents
- Sending implicit intents
- Receiving implicit intents

Recap: Intents

What is an intent?

An intent is a description of an operation to be performed.

An Intent is a messaging object used to request an action from another app component via the Android system.



What can an intent do?

An Intent can be used to:

- start an Activity
- start a Service
- deliver a Broadcast

Services and Broadcasts are covered in later lessons

Explicit and implicit intents

Explicit intent— Starts an activity of a specific class

Implicit intent—Asks system to find an activity class with a registered handler that can handle this request

Implicit Intents

Implicit intents

- An implicit intent allows you to start an activity in another app by describing an action you intend to perform, such as "share an article", "view a map", or "take a picture"
- An implicit intent specifies an action and may provide data with which to perform the action

Implicit intents

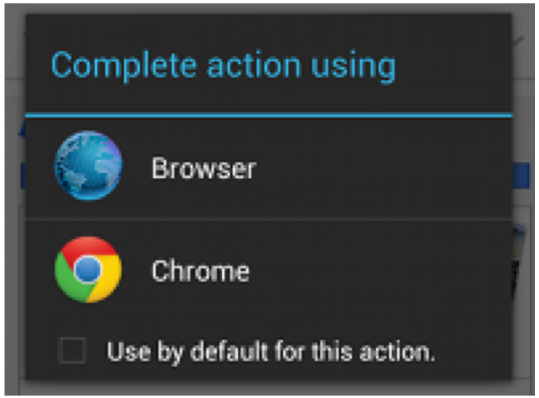
- Implicit intents do not specify the target activity class, just the intended action
- Android runtime matches the implicit intent request with registered intent handlers
- If there are multiple matches, an App Chooser will open to let the user decide

How do implicit intents work?

1. The Android Runtime keeps a list of registered Apps
2. Apps have to register via the Android Manifest
3. Runtime receives the request and looks for matches
4. Android runtime uses intent filters for matching
5. If more than one match, shows a list of possible matches and let the user choose one
6. Android runtime starts the requested activity

App Chooser

When the Android runtime finds multiple registered activities that can handle an implicit intent, it displays an App Chooser to allow the user to select the handler



Sending Implicit Intents

Sending an implicit intent

1. Create an intent for an action

```
Intent intent = new Intent(Intent.ACTION_CALL_BUTTON);
```

User has pressed Call button. Start an activity that allows them to make a call. No data is passed in or returned.

1. Start the activity

```
if (intent.resolveActivity(getPackageManager()) != null) {
    startActivity(intent);
}
```

Avoid exceptions and crashes

Before starting an implicit activity, use the package manager to check that there is a package with an activity that matches the given criteria.

```
Intent myIntent = new Intent(Intent.ACTION_CALL_BUTTON);

if (intent.resolveActivity(getPackageManager()) != null) {
    startActivity(intent);
}
```

Sending an implicit intent with data

1. Create an intent for action

```
Intent intent = new Intent(Intent.ACTION_DIAL);
```

1. Provide data as a URI

```
intent.setData(Uri.parse("tel:8005551234"));
```

1. Start the activity

```
if (intent.resolveActivity(getPackageManager()) != null) {
    startActivity(intent);
}
```

Providing the data as URI

Create an URI from a string using `Uri.parse(String uri)`

- `Uri.parse("tel:8005551234")`
- `Uri.parse("geo:0,0?q=brooklyn%20bridge%2C%20brooklyn%2C%20ny")`
- `Uri.parse("http://www.android.com");`

Uri documentation

Implicit Intents - Examples

Show a web page

```
Uri uri = Uri.parse("http://www.google.com");
Intent it = new Intent(Intent.ACTION_VIEW,uri);
startActivity(it);
```

Dial a phone number

```
Uri uri = Uri.parse("tel:8005551234");
Intent it = new Intent(Intent.ACTION_DIAL, uri);
startActivity(it);
```

Sending an implicit intent with extras

1. Create an intent for an action

```
Intent intent = new Intent(Intent.ACTION_WEB_SEARCH);
```

1. Put extras

```
String query = edittext.getText().toString();
intent.putExtra(SearchManager.QUERY, query));
```

1. Start the activity

```
if (intent.resolveActivity(getPackageManager()) != null) {
    startActivity(intent);
}
```

Category

Additional information about the kind of component to handle the intent.

- `CATEGORY_OPENABLE`
  Only allow URIs of files that are openable
- `CATEGORY_BROWSABLE`
  Only activities that can start a web browser to display data referenced by the URI

Sending an implicit intent with type and category

1. Create an intent for an action

```
Intent intent = new Intent(Intent.ACTION_CREATE_DOCUMENT);
```

1. Set mime type and category for additional information

```
intent.setType("application/pdf"); // set MIME type
intent.addCategory(Intent.CATEGORY_OPENABLE);
```

continued on next slide...

Sending an implicit intent with type and category

3. Start the activity

```
if (intent.resolveActivity(getPackageManager()) != null) {
  startActivityForResult(myIntent,ACTIVITY_REQUEST_CREATE_FILE);
}
```

4. Process returned content URI in `onActivityResult()`

Common actions for implicit intents

Common actions for implicit intents include:

- ACTION_SET_ALARM
- ACTION_IMAGE_CAPTURE
- ACTION_CREATE_DOCUMENT
- ACTION_SENDTO
- and many more

Apps that handle common actions

Common actions are usually handled by installed apps, both system apps and other apps, such as:

- Alarm Clock, Calendar, Camera, Contacts
- Email, File Storage, Maps, Music/Video
- Notes, Phone, Search, Settings
- Text Messaging and Web Browsing

➔ List of **common** actions for implicit intents

➔ List of all available actions

Receiving Implicit Intents

Register your app to receive intents

- Declare one or more intent filters for the activity in the Android manifest
- Filter announces activity's ability to accept implicit intents
- Filter puts conditions on the intents that the activity accepts

Intent filter in the Android Manifest

```
<activity android:name="ShareActivity">
  <intent-filter>
    <action android:name="android.intent.action.SEND"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="text/plain"/>
  </intent-filter>
</activity>
```

Intent filters: action and category

- `action` — Match one or more action constants
  - `android.intent.action.VIEW` – matches all intents with `ACTION_VIEW`
  - `android.intent.action.SEND` – matches all intents with `ACTION_SEND`

- `category` — additional information (list of categories)
  - `android.intent.category.BROWSABLE`—can be started by web browser
  - `android.intent.category.LAUNCHER`—Show activity as launcher icon

Intent filters: data

- `data` — Filter on data URIs, MIME type
  - `android:scheme="https"`—require URIs to be https protocol
  - `android:host="developer.android.com"`—only accept intents from specified hosts
  - `android:mimeType="text/plain"`—limit the acceptable types of documents

An activity can have multiple filters

```
<activity android:name="ShareActivity">
  <intent-filter>
    <action android:name="android.intent.action.SEND"/>
    ...
  </intent-filter>
  <intent-filter>
    <action android:name="android.intent.action.SEND_MULTIPLE"/>
    ...
  </intent-filter>
</activity>
```

An Activity can have several filters

A filter can have multiple actions & data

```
<intent-filter>
  <action android:name="android.intent.action.SEND"/>
  <action android:name="android.intent.action.SEND_MULTIPLE"/>
  <category android:name="android.intent.category.DEFAULT"/>
  <data android:mimeType="image/*"/>
  <data android:mimeType="video/*"/>
</intent-filter>
```

Learn more

Learn more

- Intent class documentation
- Uri documentation
- List of common apps that respond to implicit intents
- List of available actions
- List of categories
- Intent Filters

What's Next?

- Concept Chapter: 2.3 C Activities and Implicit Intents
- Practical: 2.3 P Start Activities with Implicit Intents

END