

Android Basics: Multiscreen Apps

Intents and Activities

Documentation: [Intent | Android Developers](https://developer.android.com/reference/android/content/Intent.html) (<https://developer.android.com/reference/android/content/Intent.html>)

- ~Intents~ are requests for an action to be performed by another app.
- Apps can specify ~intent filters~ in the `AndroidManifest.xml` file
- An intent filter watches for an intent and when received, it performs a certain action.

```
// Create the intent
Intent numbersIntent = new Intent(MainActivity.this, NumbersActivity.class);

// Start the activity specified in the Intent
startActivity(numbersIntent);
```

Documentation: [Intents and Intent Filters | Android Developers](https://developer.android.com/guide/components/intents-filters.html)
(<https://developer.android.com/guide/components/intents-filters.html>)

- ~Implicit Intents~: use these when you don't care which app component handles the intent, as long as they can handle it.
- Need an Action and Data URI
- Optional: Category, Components, Extras
- Create an object instance of the intent class and in the constructor, pass in an action string (`ACTION_SENDTO`)
- Implicit intents have a `resolveActivity` block where it handles making sure that the resolution happens

```
// Create the text message with a string
Intent sendIntent = new Intent(Intent.ACTION_SENDTO);
sendIntent.setData(Uri.parse("mailto:"));
sendIntent.putExtra(Intent.EXTRA_SUBJECT, "Just Java order for " + name);
sendIntent.putExtra(Intent.EXTRA_TEXT, priceMessage);

// Verify that the intent will resolve properly
if (sendIntent.resolveActivity(getPackageManager()) != null) {
    startActivity(sendIntent)
}
```

- ~Explicit Intents~: use these when you know what activity you want to receive your intent (usually used within an application).
 - Context and Component (usually Class/Activity)
 - ▪ Optional: Data URI

```
// Executed in an Activity, so 'this' is the Context
Intent intent = new Intent(this, NumbersActivity.class);
startActivity(intent);
```

Editing the AndroidManifest.xml File

Documentation: [App Manifest | Android Developers \(https://developer.android.com/guide/topics/manifest/manifest-intro.html\)](https://developer.android.com/guide/topics/manifest/manifest-intro.html)

The **label** attribute can be added to an activity in the `AndroidManifest.xml` file. It is what is shown to the user when they visit this activity.