

# Working with Intents

Prashanth B S<sup>1</sup>

<sup>1</sup>Department of Information Science & Engineering, Nitte Meenakshi Institute of Technology,  
Yelahanka - 560064, Bengaluru

## 1 Intent

An Intent is a messaging object you can use to request an action from another app component. Although intents facilitate communication between components in several ways, there are three fundamental use cases, They are,

- Starting an activity
- Starting a service
- Delivering a broadcast

An intent is to perform an action on the screen. It is mostly used to start activity, send broadcast receiver, start services and send message between two activities. There are two intents available in android as *Implicit* Intents and *Explicit* Intents. The implicit intents are used to invoke an activity which is part of the Android system(Ex: Dialer, Messaging Activity). The explicit intents are used to invoke an activity which is user defined. At the same time the intents are used to transfer the data between activities wrapped in the container called *Bundle*. The overall process is visualized as shown in figure 1

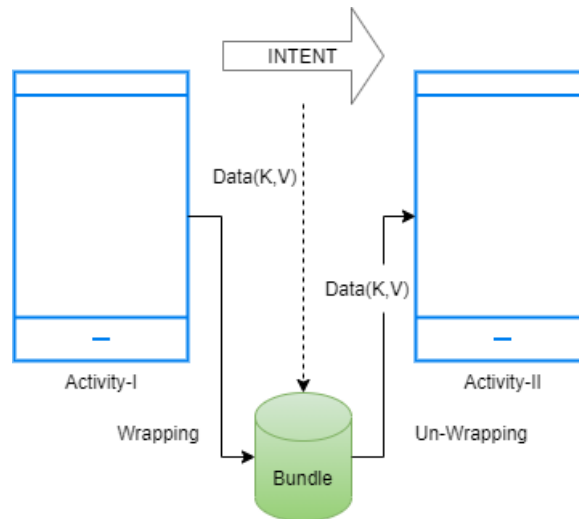


Figure 1: Intent overview

The Activity-II can be an activity either a user-defined activity or an implicit android activity, the choice differentiates the explicit and implicit intent. To create an implicit intent, the following code snippet is used,

---

```
Intent it = new Intent(Context ctx, Activity Destination_Activity);
startActivity(it);
```

---

where the destination activity is the activity that will be invoked from the current activity using context object. To Start the Activity via Intent, `startActivity(intent)` is called. The intent can carry the data with the help of bundle. To do the first create `Bundle` class object and then using that object call the function `putString(key,value)` to pass the value in the form of key-value pair. The following code snippet shows the same

```
Bundle b = new Bundle() ;
b.putString("name",name.getText().toString());
it.putExtras(b) ;
```

## 2 Exercise-I

### Create an Android application to demonstrate the working of Explicit Intent

The following are the steps followed,

- Open the `activity_main.xml`, Create a `LinearLayout` with vertical orientation. Add an `EditText` and `Button` laid out one below the other. The following code snippet shows the same,

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/name"
    android:hint="Enter your name"/>

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="CLICK"
    android:id="@+id/click"/>
```

- Create a `SecondActivity` by right clicking on the project "app" and select new->Activity->Empty Activity, set the name as `SecondActivity.java` and the xml file `activity_second.xml` as shown in figure 2.

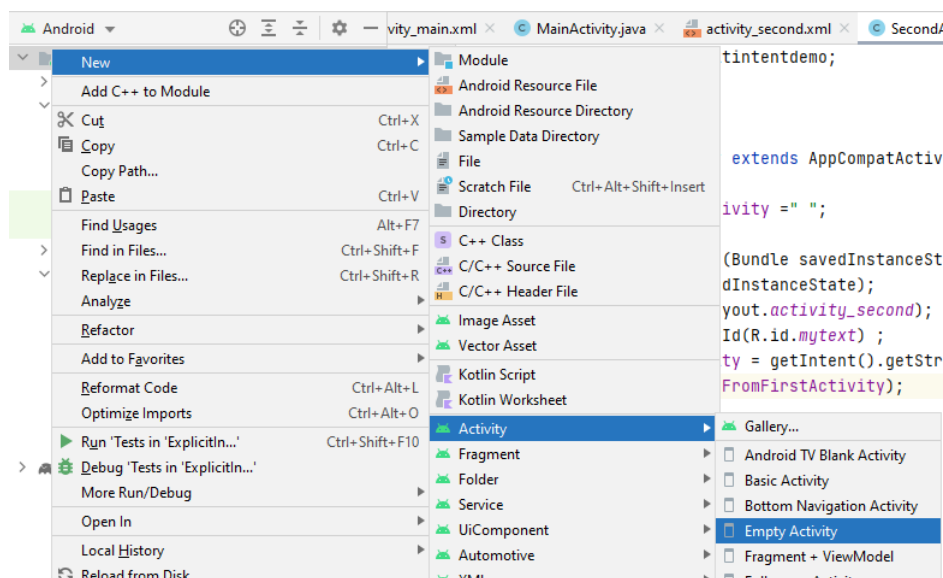


Figure 2: Creating a Second Activity

- In the activity `_second.xml`, Set the layout to `LinearLayout` and add a `textview` as a component as shown below,

---

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/mytext"
    android:text="NAME COMES HERE"
    android:textSize="40dp"
    android:textAlignment="center"
/>
```

---

- In the `MainActivity` Java file, Instantiate the java objects for the `ListView` and `Button` as follows, Instantiate the objects in the `oncreate` function too.

---

```
EditText name ;
Button click ;
// inside onCreate function
name = findViewById(R.id.name) ;
click = findViewById(R.id.click);
```

---

- Add an event `onClickListener` to the button "click". Upon click, use the `Intent` object to navigate from `MainActivity` to `SecondActivity` class with a string retrieved from the `EditText(MainActivity)` as shown below. The retrieved string is passed along with the key "name" into the `Bundle` object before passing the bundle to `Intent`. To pass the bundle to intent object use `putExtra(Bundle b)`.

---

```
click.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent it = new Intent(MainActivity.this, SecondActivity.class);
        Bundle b = new Bundle() ;
        b.putString("name",name.getText().toString());
        it.putExtras(b) ;
        startActivity(it);
    }
});
```

---

- In the `SecondActivity` define the JAVA equivalent objects for the XML `TextView` object. In the `oncreate` function, equate the XML object with JAVA object. Use `getIntent()`. `getStringExtra(Stringkey)` function to get the retrieve the string using the key("name") passed with `Intent`. Once the string is retrieved, set the text to the `TextView` object using `setText(string)` function as shown below,

---

```
TextView myName ;
String nameFromFirstActivity =" ";

//inside onCreate
myName = findViewById(R.id.mytext) ;
nameFromFirstActivity = getIntent().getStringExtra("name");
myName.setText(nameFromFirstActivity);
```

---

### 3 Exercise-II

#### Create an Android application to demonstrate the working of Implicit Intent

The app demonstrated here uses CALL\_PHONE permission to let the user to enter the Phone number and puts the number to the inbuilt dialer activity of android system. The steps are as shown below,

- Navigate to manifests directory, open the AndroidManifest XML file and add the following uses-permission(before the application tag) as shown below,

---

```
<uses-permission android:name="android.permission.CALL_PHONE"/>
```

---

- Go to the activity\_main XML, change the layout to LinearLayout with vertical orientation. Add an EditText to receive a phone number and Button for handling event click as shown below,

---

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/myphone"
    android:hint="Enter the Phone Number"
    android:layout_margin="20dp"
/>

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/call"
    android:text="CALL"
    android:layout_margin="20dp"/>
```

---

- In the MainActivity Java file, Instantiate the java objects for the EditText and Button as follows, Instantiate the objects in the *oncreate* function too.

---

```
EditText phoneNumberToCall ;
Button call ;
// inside onCreate function
call = findViewById(R.id.call) ;
phoneNumberToCall = findViewById(R.id.myphone) ;
```

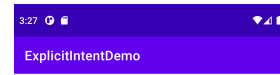
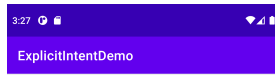
---

- Add an event *onClickListener* to the button "click". Upon click, use the Intent object to navigate from *MainActivity* to Implicit Activity that is "Dialer" in this case. The inbuilt activity can be invoked using "Intent.ACTION\_DIAL" action. The intent constructor takes two arguments they are, the *intent.ACTION*, and the URI resource(PhoneNumber as URI) which is as shown below.

---

```
call.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Uri myUri = Uri.parse("tel:"+phoneNumberToCall.getText().toString());
        Intent it = new Intent(Intent.ACTION_DIAL, myUri) ;
        startActivity(it);
    }
});
```

---



prashanth

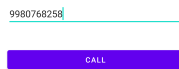
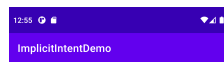


(a) Explicit Intent Front Activity

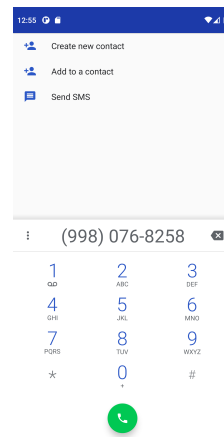


(b) Explicit Intent Second Activity

Figure 3: Explicit Intent



(a) Implicit Intent Front Activity



(b) Implicit Intent Second Activity

Figure 4: Implicit Intent

## 4 Output for Exercise-I/II

Refer the figure 3 for the output of the Explicit Intent, where in the 3a shows the front page of the App and 3b shows the SecondActivity created in the app.

Refer the figure 4 for the output of the Implicit Intent, where in the 4a shows the front page of the App and 4b shows the Dialer Activity of android system to which the Phone number is passed via Intent.