Chapter 2

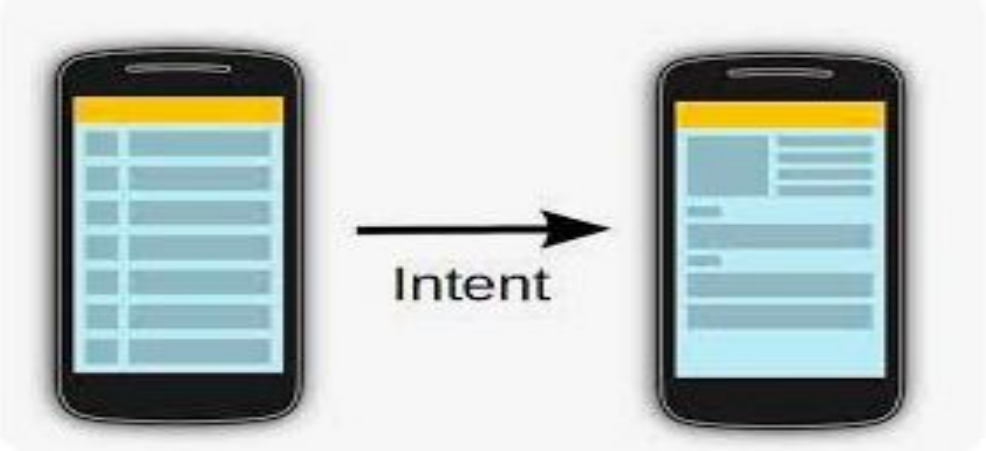
Intent

Definition

Android uses Intent for communicating between the components of an Application and also from one application to another application.

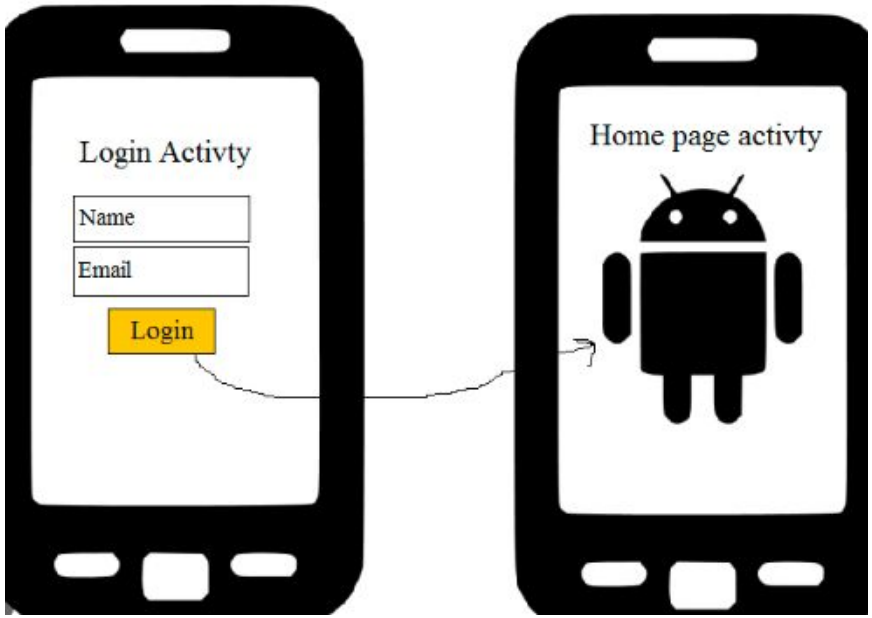
Intent are the objects which is used in android for passing the information among Activities in an Application and from one app to another also. Intent are used for communicating between the Application components and it also provides the connectivity between two apps.

Intent facilitate you to redirect your activity to another activity on occurrence of any event. By calling, startActivity() you can perform this task.



Android intents are mainly used to:

- Start the service
- Launch an activity
- Display a web page
- Display a list of contacts
- Broadcast a message
- Dial a phone call etc.



Intent Uses In Android

- **Intent for an Activity:**
  Every screen in Android application represents an activity. To start a new activity you need to pass an Intent object to startActivity() method. This Intent object helps to start a new activity and passing data to the second activity.
- **Intent for Services:**
  Services work in background of an Android application and it does not require any user Interface. Intents could be used to start a Service that performs one-time task(for example: Downloading some file) or for starting a Service you need to pass Intent to startService() method.
- **Intent for Broadcast Receivers:**
  There are various message that an app receives, these messages are called as Broadcast Receivers. (For example, a broadcast message could be initiated to intimate that the file downloading is completed and ready to use

Intents are invoked using the following options:

startActivity (intent)	launches an *Activity*
sendBroadcast (intent)	sends an intent to any interested *BroadcastReceiver* components
startService(intent) or ***bindService(intent, …)***	communicate with a background Service.

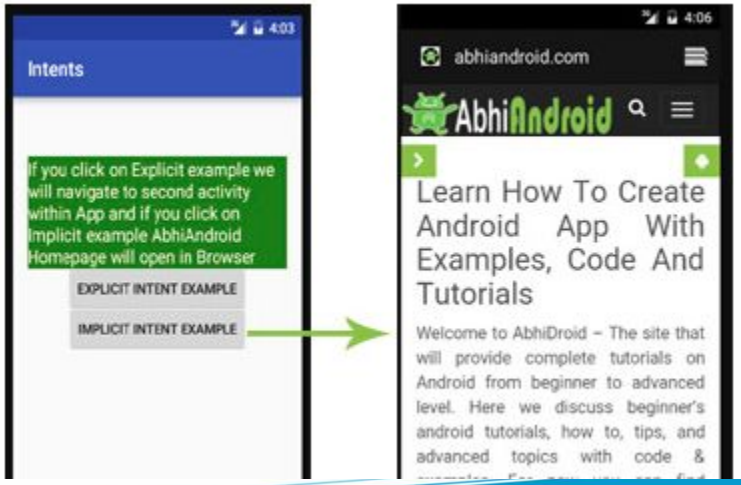
Types Of Intents

There are two types of intents in android: implicit and explicit.

1) Implicit Intent

In Implicit Intents we do need to specify the name of the component. We just specify the Action which has to be performed and further this action is handled by the component of another application.
The basic example of implicit Intent is to open any web page

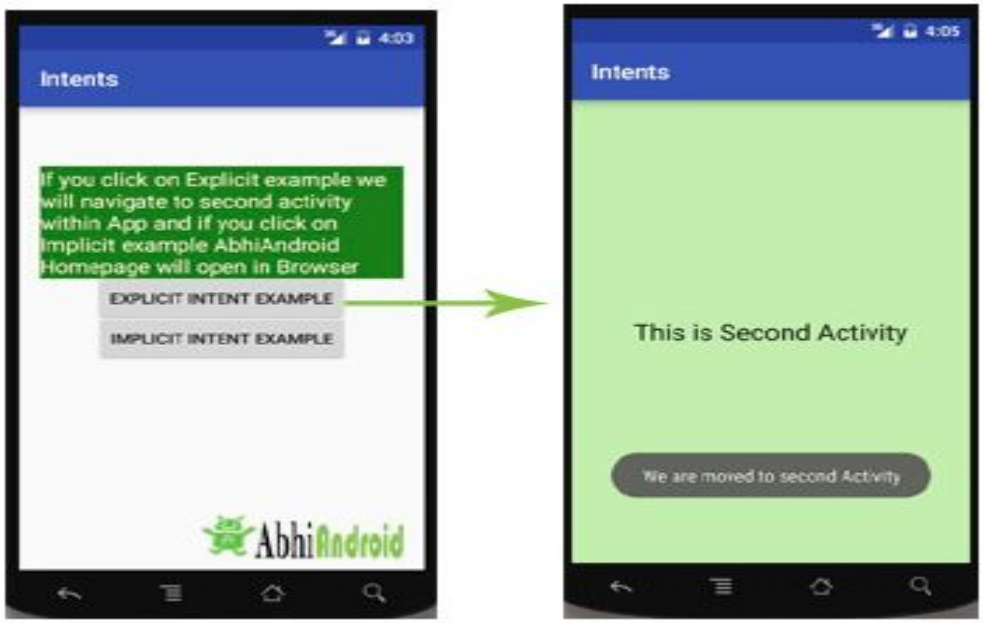
```
Intent intentObj = new Intent(Intent.ACTION_VIEW);
intentObj.setData(Uri.parse("https://www.abhiandroid.com"));
startActivity(intentObj);
```



Explicit Intent:

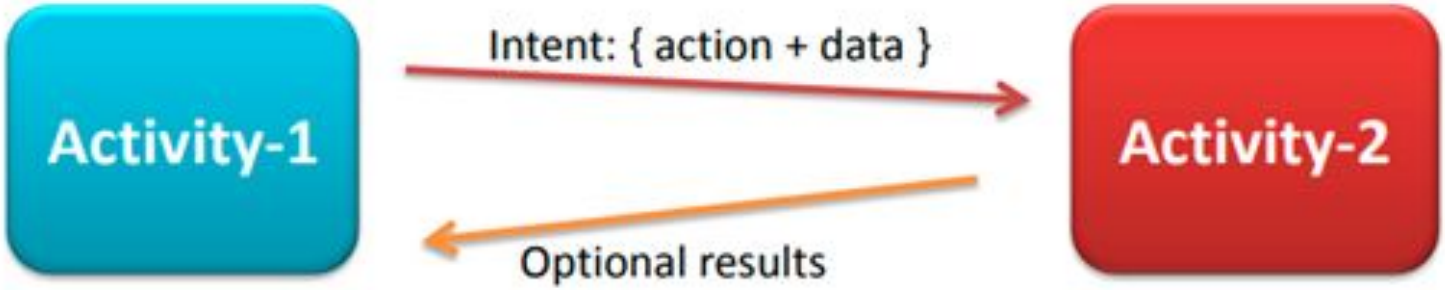
Explicit Intents are used to connect the application internally. In Explicit we use the name of component which will be affected by Intent. For Example: If we know class name then we can navigate the app from One Activity to another activity using Intent. In the similar way we can start a service to download a file in background process.

```
public void Class_Add_Btn(View view) {
    Intent i = newIntent(activity1.this,activity2.class);
     startActivity(i);
}
```



The main arguments of an Intent are:

1. **Action** The built-in action to be performed, such as `ACTION_VIEW, ACTION_EDIT, ···` or *user-created-activity*
2. **Data** The primary data to operate on, such as a phone number to be called (expressed as a **Uri**).



Typically an intent is called as follows:

```
Intent  myActivity = new Intent (action, data);

startActivity (myActivity);
```

Primary data (as an URI)
tel://
http://
sendto://

Built-in or user-created activity

Examples of action/data pairs are:

ACTION_DIAL ***tel:123***
Display the phone dialer with the given number filled in.

ACTION_VIEW ***http://www.google.com***
Show Google page in a browser view. Note how the VIEW action does what is considered the most reasonable thing for a particular URI.

ACTION_EDIT ***content://contacts/people/2***
Edit information about the person whose identifier is "2".

ACTION_VIEW ***content://contacts/people/2***
Used to start an activity to display 2-*nd* person.

ACTION_VIEW ***content://contacts/*** **people/**
Display a list of people, which the user can browse through. Selecting a particular person to view would result in a new intent

Built-in Standard Actions

List of standard actions that Intents can use for launching activities (usually through *startActivity(Intent)*.

ACTION_MAIN	ACTION_ANSWER
ACTION_VIEW	ACTION_INSERT
ACTION_ATTACH_DATA	ACTION_DELETE
ACTION_EDIT	ACTION_RUN
ACTION_PICK	ACTION_SYNC
ACTION_CHOOSER	ACTION_PICK_ACTIVITY
ACTION_GET_CONTENT	**ACTION_SEARCH**
ACTION_DIAL	**ACTION_WEB_SEARCH**
ACTION_CALL	ACTION_FACTORY_TEST
ACTION_SEND	
ACTION_SENDTO	

Example:

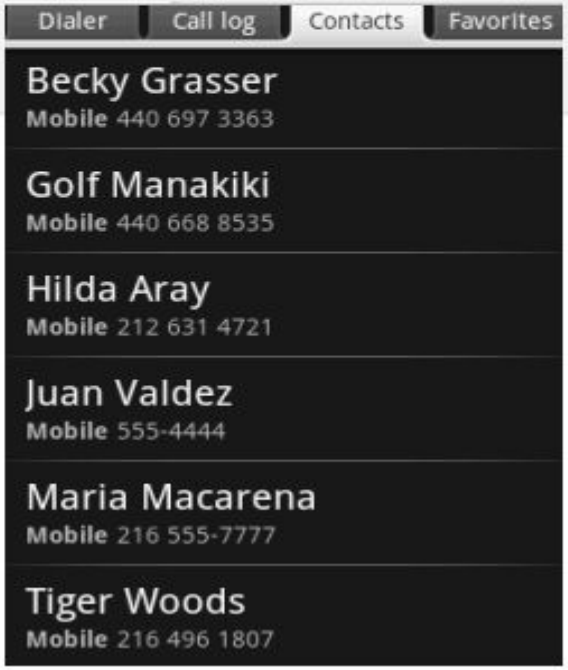
Display the phone dialer with the given number filled in.

```
Intent myActivity2 = new Intent (Intent.ACTION_DIAL,
                                 Uri.parse( "tel:555-1234"));
startActivity(myActivity2);
```

More Examples: Using Standard Actions

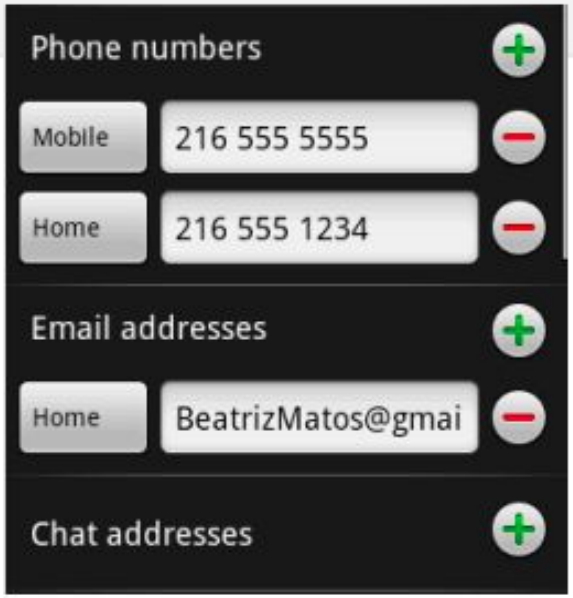
Show all your Contacts

```
String myData = "content://contacts/people/";

Intent myActivity2 = new Intent(Intent.ACTION_VIEW,
                                Uri.parse(myData));

startActivity(myActivity2);
```



Edit a Particular Contact (ID = 2)

```
String myData = "content://contacts/people/2";

Intent myActivity2 = new Intent(Intent.ACTION_EDIT,
                                Uri.parse(myData));

startActivity(myActivity2);
```



Geo Mapping an Address

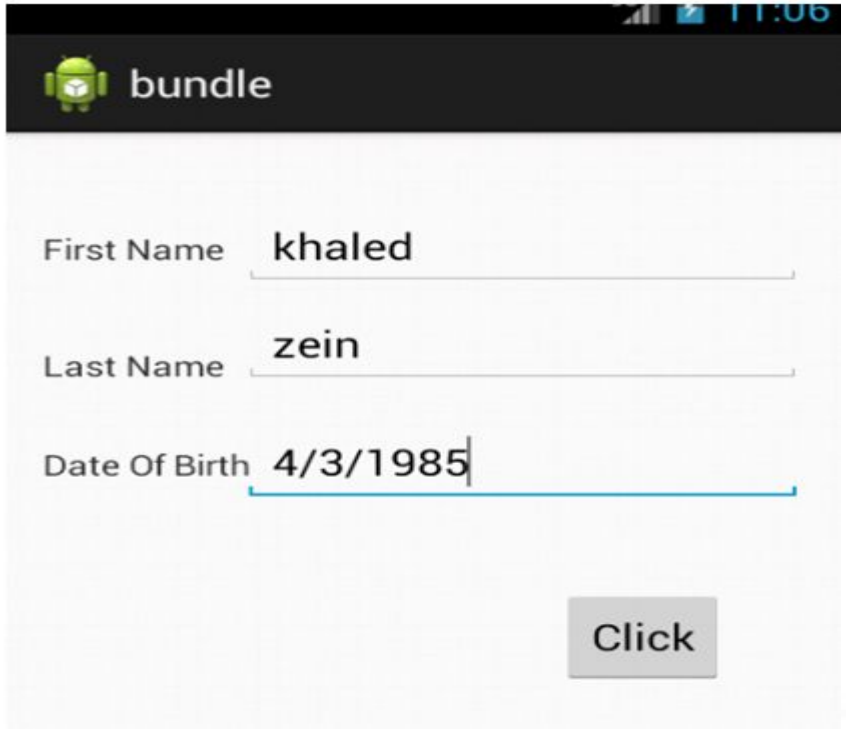

```
String geoCode =
        "geo:0,0?q=1860+east+18th+street+cleveland+oh";
Intent intent = new Intent(Intent.ACTION_VIEW,
                           Uri.parse(geoCode));
startActivity(intent);
```

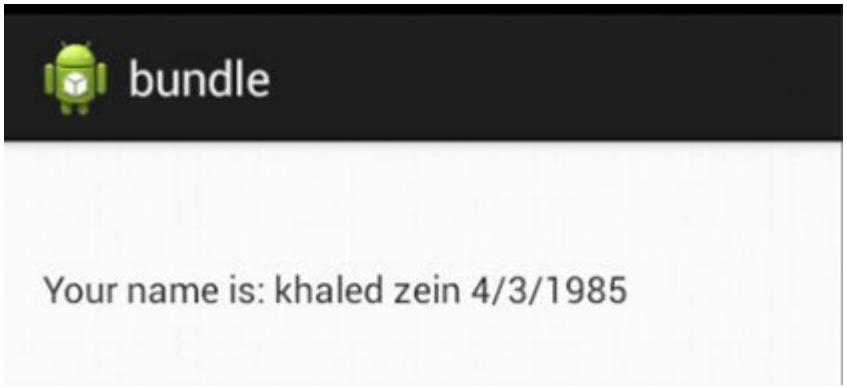
Modify the Manifest adding the following requests:

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

Bundle in Android

It is known that Intents are used in Android to pass to the data from one activity to another. But there is one another way, that can be used to pass the data from one activity to another in a better way and less code space ie by using Bundles in Android. Android Bundles are generally used for passing data from one activity to another.





Code:

Activity1.java

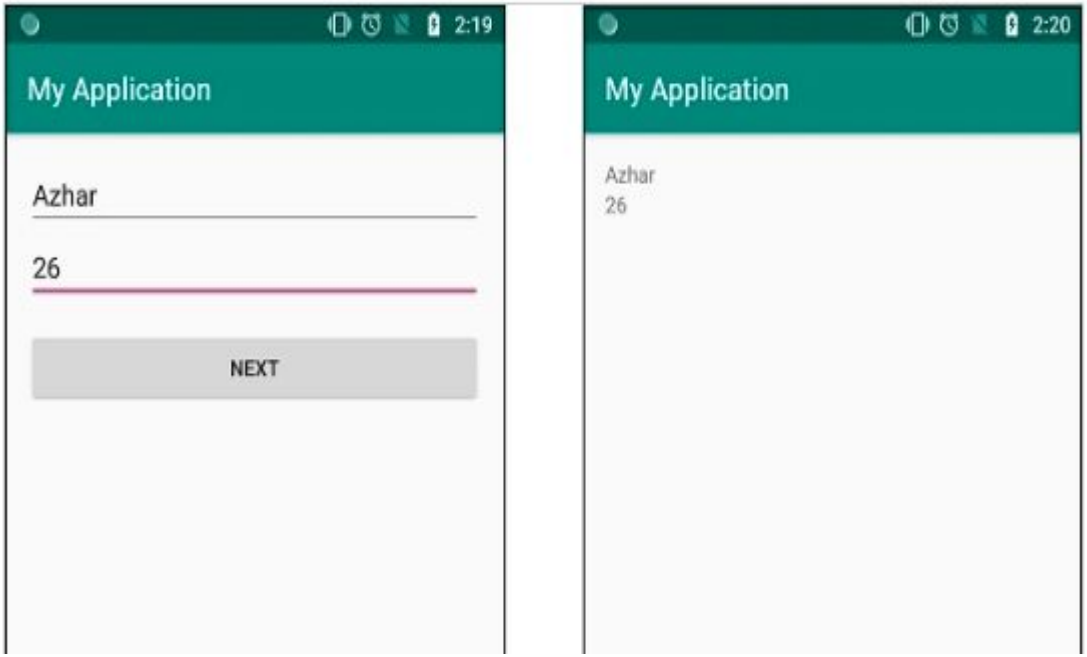
```
        Bundle b = new Bundle();
       b.putString("f",editText1
.getText().toString());
         Bundle c = new
Bundle();
       c.putString("r",editText2
.getText().toString());
       Bundle d = new Bundle();
 d.putString("s",
editText3.getText().toString);
  Intent i = new
Intent(MainActivity.this,page12.
class);

i.putExtras(b);

i.putExtras(c);

i.putExtras(d);

startActivity(i);
```

Activity2.java

```
 Intent intent = getIntent();
   String fName =
intent.getStringExtra("f");
   String lName =
intent.getStringExtra("r");
 String birth =
intent.getStringExtra("s");
 txt_view1.setText("Your name
is: " + fName + "
" + lName+" "+ birth);
```

Example:

How to pass multiple data from one activity to another in Android?



Activity1.java

```
etName = findViewById(R.id.etName);
etAge = findViewById(R.id.etAge);
Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String name = etName.getText().toString().trim();
        String age = etAge.getText().toString().trim();
        Bundle bundle = new Bundle();
        bundle.putString("name", name);
        bundle.putString("age", age);
        Intent intent = new Intent(MainActivity.this, SecondActivity.class);
        intent.putExtras(bundle);
        startActivity(intent);
    }
});
```

Activity2.java

```
Bundle bundle = getIntent().getExtras();
if (bundle ! = null) {
    String name = bundle.getString("name");
    String age = bundle.getString("age");
    TextView tvName = findViewById(R.id.tvName);
    TextView tvAge = findViewById(R.id.tvAge);
    tvName.setText(name);
    tvAge.setText(age);
}
```

Intent Filters

An intent filter is an instance of the IntentFilter class. Intent filters are helpful while using implicit intents, It is not going to handle in java code, we have to set it up in AndroidManifest.xml. Android must know what kind of intent it is launching so intent filters give the information to android about intent and actions.

The intent filter specifies the types of intents that an activity, service, or broadcast receiver can respond.

Before launching intent, android going to do action test, category test and data test.

action - we use this property to define that the activity can perform **SEND** action.

category - we included the **DEFAULT** category for this activity to be able to receive implicit intents.

data - the type of data the activity can send.

Intent Filters

<action>

ACTION_VIEW: Use this action in intent with startActivity() when you have some information that activity can show to the user like showing an image in a gallery app or an address to view in a map app

ACTION_SEND: You should use this in intent with startActivity() when you have some data that the user can share through another app, such as an email app or social sharing app.

<category>

CATEGORY_BROWSABLE: The target activity allows itself to be started by a web browser to display data referenced by a link.

<data>

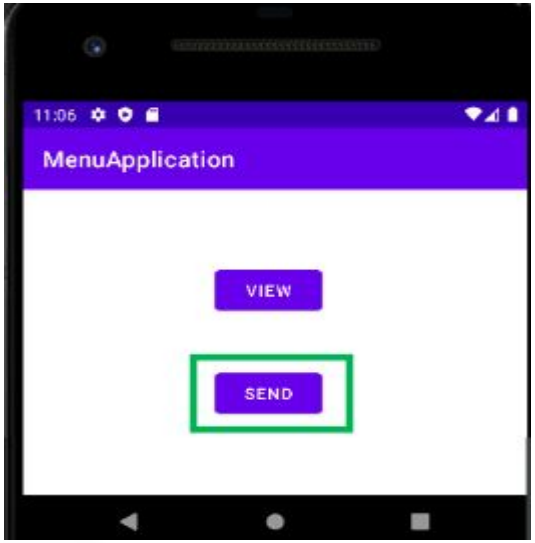
Adds a data specification to an intent filter. The specification can be just a data type, just a URI, or both a data type and a URI.

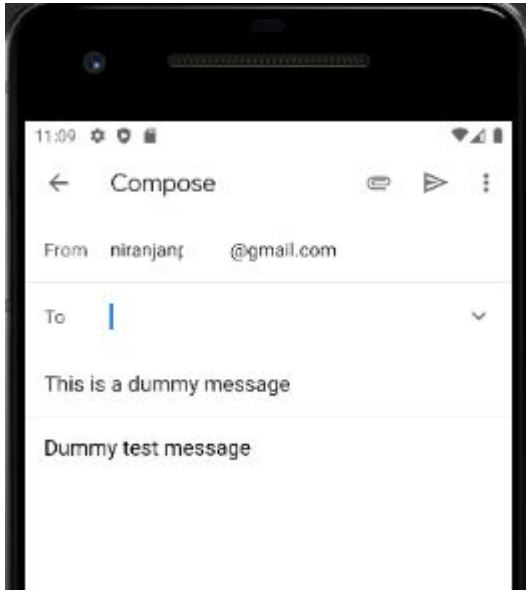
Intent Filters

For example, here's an activity declaration with an intent filter to receive an ACTION_SEND intent when the data type is text:

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

<!--VIEW INTENT FILTER-->
<intent-filter>
    <action android:name="android.intent.action.VIEW"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <category android:name="android.intent.category.BROWSABLE"/>
    <data android:scheme="http"/>
</intent-filter>
```

```
// send button on click listener
sendButton.setOnClickListener {
    var intent = Intent(Intent.ACTION_SEND) // intent
    intent.type = "text/plain"
    intent.putExtra(Intent.EXTRA_EMAIL, "niranjantest@gmail.com")
    intent.putExtra(Intent.EXTRA_SUBJECT, "This is a dummy message")
    intent.putExtra(Intent.EXTRA_TEXT, "Dummy test message")
    startActivity(intent)
}
```






ANDROID
THANK
YOU!