

## ANDROID INTENTS

An Intent is a messaging object you can use to request an action from another app component.

Three fundamental use cases:

### 1. Starting an activity

An Activity represents a single screen in an app. You can start a new instance of an Activity by passing an Intent to `startActivity()`. The Intent describes the activity to start and carries any necessary data.

If you want to receive a result from the activity when it finishes, call `startActivityForResult()`. Your activity receives the result as a separate Intent object in your activity's `onActivityResult()` callback.

### 2. Starting a service

A Service is a component that performs operations in the background without a user interface.

### 3. Delivering a broadcast

A broadcast is a message that any app can receive. The system delivers various broadcasts for system events, such as when the system boots up or the device starts charging. You can deliver a broadcast to other apps by passing an Intent to `sendBroadcast()` or `sendOrderedBroadcast()`.

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.

There are two types of intents:

- **Explicit intents** specify which application will satisfy the intent, by supplying either the target app's package name or a fully-qualified component class name. You'll typically use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start. For example, you might start a new activity within your app in response to a user action, or start a service to download a file in the background.
- **Implicit intents** do not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it. For example, if you want to show the user a location on a map, you can use an implicit intent to request that another capable app show a specified location on a map.

### Example explicit intent

An explicit intent is one that you use to launch a specific app component, such as a particular activity or service in your app. To create an explicit intent, define the component name for the Intent object—all other intent properties are optional.

```
Intent i = new Intent(MainActivity.this, ActivityTwo.class);
startActivity(i);
```

### Example implicit intent

An implicit intent specifies an action that can invoke any app on the device able to perform the action. Using an implicit intent is useful when your app cannot perform the action, but other apps probably can and you'd like the user to pick which app to use.

For example, if you have content that you want the user to share with other people, create an intent with the ACTION\_SEND action and add extras that specify the content to share. When you call startActivity() with that intent, the user can pick an app through which to share the content.

```
// Create the text message with a string
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, textMessage);
sendIntent.setType("text/plain");

startActivity(sendIntent);
```

```
Intent sendIntent = new Intent(Intent.ACTION_SEND);
// Always use string resources for UI text.
// This says something like "Share this photo with"
String title = "Some Title";
// Create intent to show the chooser dialog
Intent chooser = Intent.createChooser(sendIntent, title);
startActivity(chooser)
```

### Dial a phone number

```
Uri uri = Uri.parse("tel:1234567890");
Intent it = new Intent(Intent.ACTION_DIAL, uri);
startActivity(it);
```

### Example 1:

Develop an application for sending email.

#### XML Code:

activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="ty.practical6.MainActivity">

    <Button
        android:id="@+id/btnSendEmail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="92dp"
        android:onClick="sendEmail"
        android:text="Compose Email"
        android:layout_alignTop="@+id/txtMessage"
        android:layout_centerHorizontal="true" />

    <EditText
        android:id="@+id/txtMessage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Message"
        android:inputType="textMultiLine"
        android:singleLine="true"
        android:layout_marginTop="48dp"
        android:layout_below="@+id/txtSubject"
        android:layout_centerHorizontal="true" />

    <EditText
        android:id="@+id/txtEmailTo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="To"
        android:inputType="textEmailAddress"
        android:layout_marginTop="22dp"
        android:layout_alignParentTop="true"
        android:layout_alignStart="@+id/txtSubject" />

    <EditText
        android:id="@+id/txtSubject"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignStart="@+id/txtMessage"
```

```
    android:layout_below="@+id/txtEmailTo"
    android:layout_marginTop="43dp"
    android:ems="10"
    android:hint="Subject"
    android:inputType="text" />
```

</RelativeLayout>

#### Source Code:

MainActivity.java

```
package ty.practical6;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void sendEmail(View v) {

        EditText txtEmailTo = (EditText) findViewById(R.id.txtEmailTo);
        EditText txtSubject = (EditText) findViewById(R.id.txtSubject);
        EditText txtMessage = (EditText) findViewById(R.id.txtMessage);

        String[] TO = {txtEmailTo.getText().toString()};
        String[] CC = {" "};
        String subject = txtSubject.getText().toString();
        String msg = txtMessage.getText().toString();

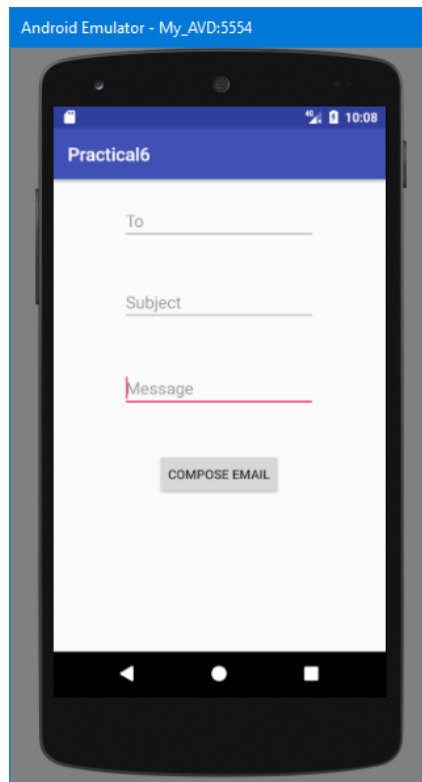
        Intent emailIntent = new Intent(Intent.ACTION_SEND);

        emailIntent.setData(Uri.parse("mailto:"));
        emailIntent.setType("text/plain");
        emailIntent.putExtra(Intent.EXTRA_EMAIL, TO);
        emailIntent.putExtra(Intent.EXTRA_CC, CC);
        emailIntent.putExtra(Intent.EXTRA_SUBJECT, subject);
        emailIntent.putExtra(Intent.EXTRA_TEXT, msg);
```

```

    try {
        startActivity(Intent.createChooser(emailIntent, "Send mail..."));
        finish();
    } catch (android.content.ActivityNotFoundException ex) {
        Toast.makeText(MainActivity.this, "No email client app installed.",
            Toast.LENGTH_SHORT).show();
    }
}
}
}
}
Output:

```



On clicking COMPOSE EMAIL button a list of apps will be displayed select a relevant email client app e.g. Gmail and the contents given as input here will be passed to Gmail app's email compose screen.

## Example 2:

Develop an application for working with device camera.

### XML Code:

activity\_main.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="ty.practical8.MainActivity">

```

```

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:onClick="takePhotos"
    android:text="Take a Photo"></Button>

```

```

<ImageView
    android:id="@+id/imageView1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true">
</ImageView>

```

</RelativeLayout>

#### Source Code:

MainActivity.java

```

package ty.practical8;

import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_CODE = 1;
    ImageView imageView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        imageView = (ImageView) this.findViewById(R.id.imageView1);
    }

    public void takePhotos(View v){
        Intent cameraIntent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(cameraIntent, REQUEST_CODE);
    }
}

```

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == REQUEST_CODE && resultCode == RESULT_OK && data != null) {
        Bitmap photo = (Bitmap) data.getExtras().get("data");
        imageView.setImageBitmap(photo);
    }
}
}

```

Output:

