Lab 2: Intent

Background

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:

- **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. For more information, see the Activities guide.
- **Starting a service**
  A Service is a component that performs operations in the background without a user interface. With Android 5.0 (API level 21) and later, you can start a service with JobScheduler. For more information about JobScheduler, see its API-reference documentation.
  For versions earlier than Android 5.0 (API level 21), you can start a service by using methods of the Service class. You can start a service to perform a one-time operation (such as downloading a file) by passing an Intent to startService(). The Intent describes the service to start and carries any necessary data.
  If the service is designed with a client-server interface, you can bind to the service from another component by passing an Intent to bindService(). For more information, see the Services guide.
- **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().

The rest of this page explains how intents work and how to use them. For related information, see Interacting with Other Apps and Sharing Content.

Intent type

- **Implicit intent**
  It 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.

- **Explicit intent**
  It 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.

Experiments

- Implicit intent
- Explicit intent
- Explicit intent with value

Activity Classes

i) MainActivity.java

```
package com.example.lab2_intent;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity  implements View.OnClickListener{
   EditText tvPassExplicit;
   Intent i;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Button implicit = findViewById(R.id.btnImplicit);
      Button explicit = findViewById(R.id.btnExplicit);
      Button passExplicit = findViewById(R.id.btnPassExplicit);
      tvPassExplicit  = findViewById(R.id.tvPassExplicit);
      implicit.setOnClickListener(this);
      explicit.setOnClickListener(this);
      passExplicit.setOnClickListener(this);
   }
   @Override
   public void onClick(View view) {
      if(view.getId()==R.id.btnImplicit){
         String url = tvPassExplicit.getText().toString();
         if (url != null && !url.equals("")) {
            if (!(url.startsWith("https://" )) && !(url.startsWith("http//" ))) {
               url = "http://www.google.com/search?q=" + url;
            }
            i=new Intent(Intent.ACTION_VIEW);
```

```
                i.setData(Uri.parse(url));
                startActivity(i);
            } else {
                Toast.makeText(MainActivity.this, "Empty URL", Toast.LENGTH_SHORT
).show();
            }
        }
        if(view.getId()==R.id.btnExplicit){
            i = new Intent(getApplicationContext(), ExplicitActivity.class);
            startActivity(i);
        }
        if(view.getId()==R.id.btnPassExplicit){
            i = new Intent(MainActivity.this, ValuePassExplicitActivity.class);
            i.putExtra("text", tvPassExplicit.getText().toString());
            startActivity(i);
        }
    }
}
```

ii) ExplicitActivity.java

```
package com.example.lab2_intent;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class ExplicitActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_explicit);
    }
}
```

iii) ValuePassExplicitActivity.java

```
package com.example.lab2_intent;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
```

```
public class ValuePassExplicitActivity extends AppCompatActivity {
    TextView txtView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_value_pass_explicit);
        txtView = (TextView) findViewById(R.id.receivedText);
        txtView.setText(getIntent().getExtras().getString("text"));
    }
}
```

Layouts Files

i) activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Lab:2 Intent"
        android:id="@+id/header"
        android:textSize="40sp"
        android:textColor="#2196F3"
        android:textStyle="bold"
        android:layout_centerHorizontal="true"
        android:layout_margin="40dp"
        />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/header"
        android:id="@+id/tvPassExplicit"
        android:hint="Input Text"
        android:layout_marginBottom="30dp"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:text="Implicit "
        android:id="@+id/btnImplicit"
        android:layout_below="@+id/tvPassExplicit"
        android:layout_centerHorizontal="true"
        android:layout_margin="30dp"
        />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Explicit"
        android:layout_below="@+id/btnImplicit"
        android:id="@+id/btnExplicit"
        android:layout_centerHorizontal="true"
        android:layout_margin="30dp"
        />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pass value to explicit "
        android:id="@+id/btnPassExplicit"
        android:layout_below="@+id/btnExplicit"
        android:layout_centerHorizontal="true"
        android:layout_margin="30dp"
        />
</RelativeLayout>
```

ii) activity_explicit.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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=".ExplicitActivity">
</androidx.constraintlayout.widget.ConstraintLayout>
```

iii) activity_value_pass_explicit.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/receivedText"
        android:textSize="30sp"
        android:textColor="#2196F3"
        android:textStyle="bold"
        android:layout_centerHorizontal="true"
        android:layout_margin="40dp"
        />
</RelativeLayout>
```

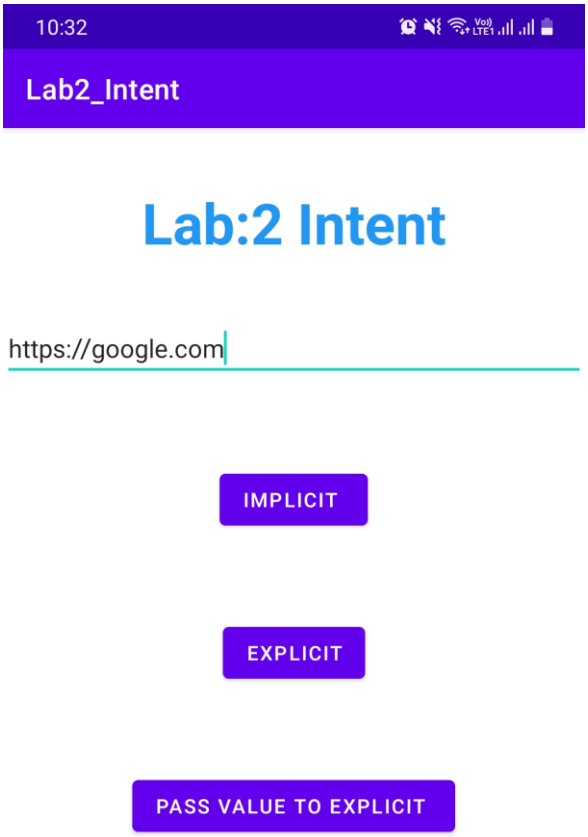
Manifest code

i) AndroidManifest.xml

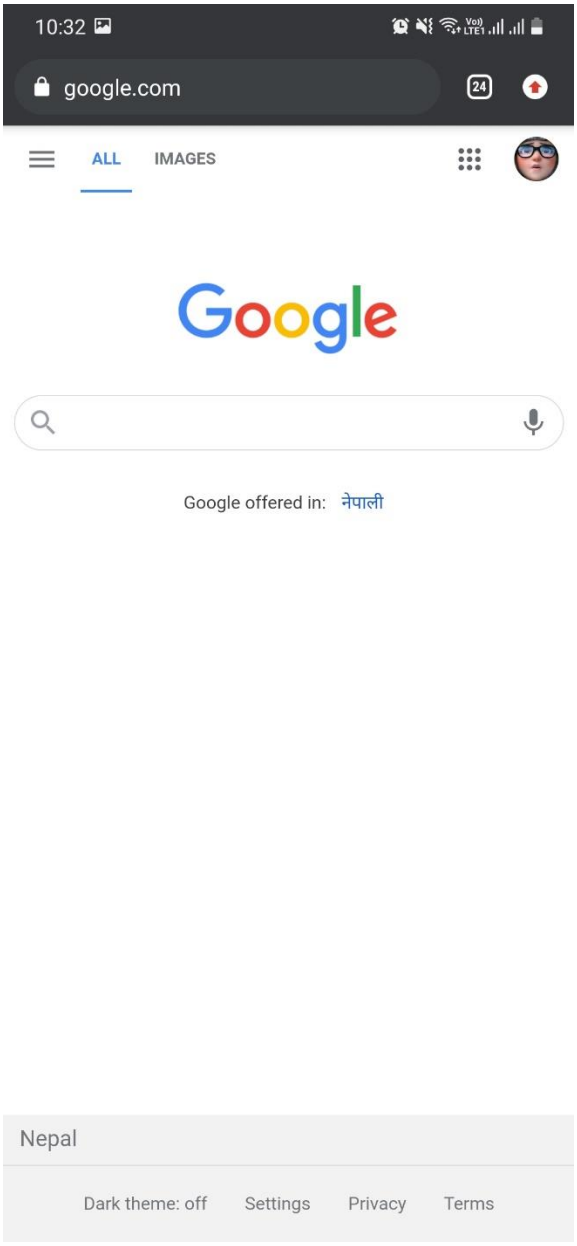
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.lab2_intent">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Lab2_Intent">
        <activity
            android:name=".ValuePassExplicitActivity"
            android:exported="false" />
        <activity
            android:name=".ExplicitActivity"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
```

```
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Output Screenshot



Implicit intent



Explicit intent

Explicit intent with value