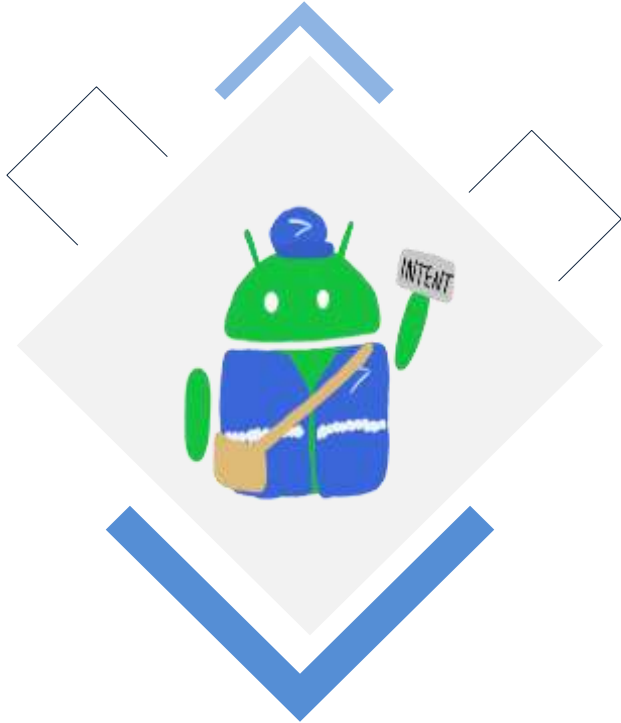




SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approved by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai



INTENTS

Course: Mobile Application Development

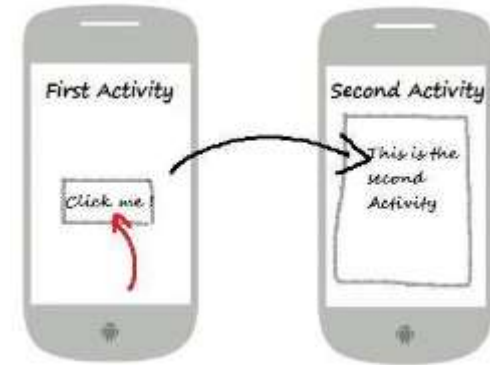
Unit : III – Building Blocks of Mobile Apps - II

Class / Semester: II MCA / III Semester

Department of MCA



- ❑ It is the **message** (In the form of OBJECT) that is passed between components such as activities, content providers, broadcast receivers, services etc.
- ❑ It is generally used with `startActivity()` method to invoke activity, broadcast receivers etc.
- ❑ Intent are used for communicating between the Apps
- ❑ 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.





- ❑ Two important parts of a Intent is "**action**" and "**data**"
- ❑ “action” is the verb, like
 - ACTION_VIEW - to bring up a viewer for the resource
 - ACTION_EDIT - to edit the resource
 - ACTION_PICK - to choose an available item given a URI
- ❑ “data” is a URI like
 - content://contact/people/1 - representing a contact in the contacts database in your device
- ❑ other criteria you can place inside an intent Start the service
 - Category – it may be LAUNCHER, DEFAULT or ALTERNATIVE
 - MIME type - indicating the type of resource you want to operate on
 - A component -the class of the activity that is supposed to receive this intent.
 - Extras- which is a Bundle of other information you want to pass along to the receiver with the intent,



- ❑ **Extras:** You can add extra data to an Intent in the form of key-value pairs and this extra information can be passed from one Activity to the other
- ❑ ***putExtra()*** is used to add some extra data to the Intents which has key & value parameters
- ❑ Intent filter in the manifest file
- ❑ **Intent Filter** is an expression in the app's **manifest** file (**ActivityMainfest.xml**) and it is used to specify the type of intents that the component would like to receive
- ❑ Intent filter in the manifest file

```
<activity android:name=".MainActivity">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
    <data android:mimeType="text/plain"/>
  </intent-filter>
</activity>
```



Implicit Intents

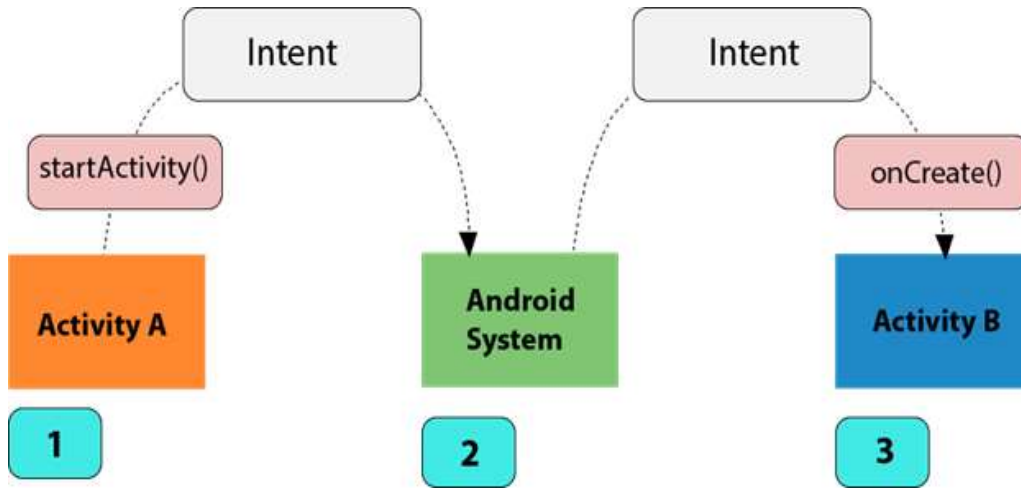
- used to connect the application internally
- we use the name of component which will be affected by Intent

```
Intent intent = new Intent(getApplicationContext(),  
SecondActivity.class); startActivity(intent);
```

Explicit Intents

- need to specify the name of the component
- just specify the Action which has to be performed

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



How an **Implicit** intent is used when starting an activity?

1. *Activity A* creates an Intent with an action description and passes it to **startActivity()**
2. Android System searches all apps for an intent filter that matches the intent. When a match is found,
3. the system starts the matching activity (*Activity B*) by invoking its **onCreate()** method and passing it the Intent

When the Intent (**Explicit**) object names a specific activity component explicitly, the system immediately starts that component



```
<activity android:name=".MainActivity">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
    <data android:mimeType="text/plain"/>
  </intent-filter>
</activity>
```



- ❑ Create a new application and open activity_main.xml in the res folder



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.tutlane.intents.MainActivity">
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/urlText"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:ems="10" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnNavigate"
        android:layout_below="@+id/urlText"
        android:text="Navigate"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```




- ❑ Then open the main activity file **MainActivity.java** from java folder

```
package com.tutlane.intents;
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final EditText editText = (EditText)findViewById(R.id.urlText);
        Button btn = (Button) findViewById(R.id.btnNavigate);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String url = editText.getText().toString();
                Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
                startActivity(intent);
            }
        });
    }
}
```



Thank You

