Intent in Android

Intent Tutorial in Android with Example And Types

Android uses [Intent](https://abhiandroid.com/programming/intent-in-android/) for communicating between the components of an Application and also from one application to another application.

[Intent](https://abhiandroid.com/programming/intent-in-android/) 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](https://abhiandroid.com/programming/intent-in-android/) are used for communicating between the Application components and it also provides the connectivity between two apps.

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

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

startActivity(intent);

In the above example, foreground activity is getting redirected to another activity i.e. SecondActivity.[java](https://abhiandroid.com/java/). getApplicationContext() returns the context for your foreground activity.

#### ****Types of Intents:****

Intent are of two types: Explicit Intent and Implicit Intent

**Explicit Intent:**

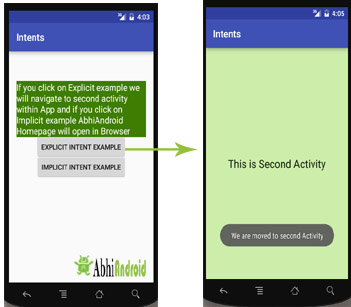
* Explicit Intents are used to connect the application internally.
* In Explicit we use the name of component which will be affected by Intent. For Example: If we know class name then we can navigate the app from One Activity to another activity using Intent. In the similar way we can start a service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

Explicit Intent work internally within an application to perform navigation and data transfer. The below given code snippet will help you understand the concept of Explicit Intents

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

startActivity(intent);

Here SecondActivity is the [JAVA](https://abhiandroid.com/java/) class name where the activity will now be navigated. Example with code in the end of this post will make it more clear.



**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

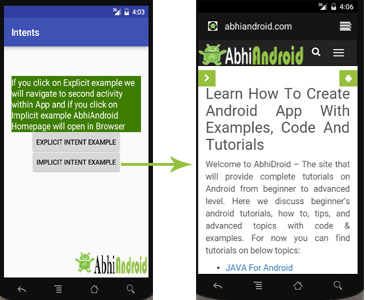
Let’s take an example to understand Implicit Intents more clearly. We have to open a website using intent in your application. See the code snippet given below

Intent intentObj = new Intent(Intent.ACTION\_VIEW);

intentObj.setData(Uri.parse("https://www.shivatutorials.com"));

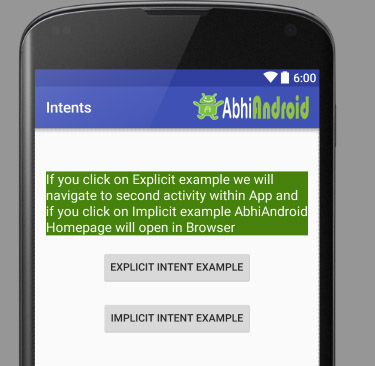
startActivity(intentObj);

Unlike Explicit Intent you do not use any class name to pass through Intent(). In this example we has just specified an action. Now when we will run this code then Android will automatically start your web browser and it will open Shivatutorials home page.



#### ****Intent Example In Android:****

Let’s implement Intent for a very basic use. In the below example we will Navigate from one Activity to another and open a web homepage of Shivatutorials using Intent. **The example will show you both implicit and explicit Intent together. Below is the final output:**



Create a project in [Android Studio](https://abhiandroid.com/androidstudio/) and named it “Intents”. Make an activity, which would consists [Java](https://abhiandroid.com/java/) file; MainActivity.java and an [xml](https://abhiandroid.com/ui/xml/) file for User interface which would be activity\_main.[xml](https://abhiandroid.com/ui/xml/)

**Step 1: Let’s design the UI of activity\_main.xml:**

* First design the [text view](https://abhiandroid.com/ui/textview/) displaying basic details of the App
* Second design the two [button](https://abhiandroid.com/ui/button/) of Explicit Intent Example and Implicit Intent Example

**Below is the complete code of activity\_main.xml**

<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" android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin" tools:context=".MainActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceMedium"

android:text="If you click on Explicit example we will navigate to second activity within App and if you click on Implicit example Shivatutorials Homepage will open in Browser"

android:id="@+id/textView2"

android:clickable="false"

android:layout\_alignParentTop="true"

android:layout\_alignParentStart="true"

android:layout\_marginTop="42dp"

android:background="#3e7d02"

android:textColor="#ffffff" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Explicit Intent Example"

android:id="@+id/explicit\_Intent"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="147dp" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Implicit Intent Example"

android:id="@+id/implicit\_Intent"

android:layout\_centerVertical="true"

android:layout\_centerHorizontal="true" />

</RelativeLayout>

**Step 2: Design the UI of second activity activity\_second.xml**

Now lets design UI of another activity where user will navigate after he click on Explicit Example [button](https://abhiandroid.com/ui/button/). Go to layout folder, create a new activity and name it activity\_second.[xml](https://abhiandroid.com/ui/xml/).

* In this activity we will simply use [TextView](https://abhiandroid.com/ui/textview/" \o "TextView" \t "_self) to tell user he is now on second activity.

**Below is the complete code of activity\_second.xml**

<?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" android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:background="#CCEEAA"

tools:context="com.example.android.intents.SecondActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge"

android:text="This is Second Activity"

android:id="@+id/textView"

android:layout\_centerVertical="true"

android:layout\_centerHorizontal="true" />

</RelativeLayout>

**Step 3: Implement onClick event for Implicit And Explicit Button inside MainActivity.java**

Now we will use setOnClickListener() method to implement OnClick event on both the [button](https://abhiandroid.com/ui/button/). Implicit button will open Shivatutorials.com homepage in browser and Explicit button will move to SecondActivity.java.

**Below is the complete code of MainActivity.java**

package com.example.android.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;

public class MainActivity extends AppCompatActivity {

Button explicit\_btn, implicit\_btn;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

explicit\_btn = (Button)findViewById(R.id.explicit\_Intent);

implicit\_btn = (Button) findViewById(R.id.implicit\_Intent);

//implement Onclick event for Explicit Intent

explicit\_btn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(getBaseContext(), SecondActivity.class);

startActivity(intent);

}

});

//implement onClick event for Implicit Intent

implicit\_btn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(Intent.ACTION\_VIEW);

intent.setData(Uri.parse("https://www.shivatutorials.com"));

startActivity(intent);

}

});

}

}

**Step 4: Create A New JAVA class name SecondActivity**

Now we need to create another SecondActivity.java which will simply open the layout of activity\_second.xml . Also we will use [Toast](https://abhiandroid.com/programming/custom-toast-tutorial-example.html/) to display message that he is on second activity.

**Below is the complete code of SecondActivity.java:**

package com.example.android.intents;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.Toast;

public class SecondActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_second);

Toast.makeText(getApplicationContext(), "We are moved to second Activity",Toast.LENGTH\_LONG).show();

}

}

**Step 5: Manifest file:**

Make sure Manifest file has both the MainActivity and SecondActivity listed it. Also here MainActivity is our main activity which will be launched first. So make sure intent-filter is correctly added just below MainActivity.

**Below is the code of Manifest file:**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.android.intents" >

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:supportsRtl="true"

android:theme="@style/AppTheme" >

<activity android:name=".MainActivity" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity android:name=".SecondActivity" >

</activity>

</application>

</manifest>