

Topic: Intents & Navigation

What is an Intent?

An **Intent** in Android is a message object that allows components (like Activities, Services, and Broadcast Receivers) to communicate with each other.

It is used to:

- Start a new activity
- Start a service
- Deliver a broadcast message
- Pass data between components

In simple terms, **Intents help navigate from one screen (Activity) to another.**

Types of Intents

Explicit Intent

- Used when you know **which activity or component** you want to open.
- Usually used **within the same app**.
- Example: Opening another screen (Activity) in your app.

```
val intent = Intent(this, SecondActivity::class.java)
startActivity(intent)
```

Implicit Intent

- Used when you want to perform a general action, and **let Android decide** which app can handle it.
- Example: Opening a web page, sharing text, or dialing a phone number.

```
val intent = Intent(Intent.ACTION_VIEW)
intent.data = Uri.parse("https://www.google.com")
startActivity(intent)
```

Passing Data Between Activities

When you use **Explicit Intents**, you can send data to another Activity using:

```
intent.putExtra("keyName", value)
And receive it in the next Activity using:
val value = intent.getStringExtra("keyName")
```

Program: Two-Screen App using Intents

Create an app with two screens:

- **MainActivity:** Takes user input (name)
- **SecondActivity:** Displays the received name

Step-by-Step Implementation

Step 1: Create Two Activities

- MainActivity.kt
- SecondActivity.kt

Each with its own layout (activity_main.xml, activity_second.xml).

Step 2: activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="20dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:id="@+id/etName"
        android:hint="Enter your name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

    <Button
        android:id="@+id/btnSend"
        android:text="Go to Next Screen"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

Step 3: MainActivity.kt

```
package com.example.intentsdemo

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val etName = findViewById<EditText>(R.id.etName)
        val btnSend = findViewById<Button>(R.id.btnSend)

        btnSend.setOnClickListener {
            val name = etName.text.toString()

            // Explicit Intent to open SecondActivity
```

```
        val intent = Intent(this, SecondActivity::class.java)
        intent.putExtra("username", name)
        startActivity(intent)
    }
}
```

Step 4: activity_second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="20dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/tvResult"
        android:textSize="22sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Hello!" />
</LinearLayout>
```

Step 5: SecondActivity.kt

```
package com.example.intentsdemo

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

class SecondActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_second)

        val tvResult = findViewById<TextView>(R.id.tvResult)
        val name = intent.getStringExtra("username")

        tvResult.text = "Welcome, $name!"
    }
}
```

Step 6: AndroidManifest.xml

Make sure both activities are declared:

```
<application
    android:allowBackup="true"
    android:theme="@style/Theme.AppCompat.Light.DarkActionBar">

    <activity android:name=".SecondActivity" />
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

</application>
```

Output

- Screen 1: User enters their name and taps the button.
- Screen 2: Displays “Welcome, [Name]!”

Note:

You can add another button to open Google:

```
val browserIntent = Intent(Intent.ACTION_VIEW, Uri.parse("https://www.google.com"))
startActivity(browserIntent)
```

Additional inputs

Checklist: Files and Settings for Intent Demo App

You only need **4 things** properly configured:

AndroidManifest.xml

Open app/manifests/AndroidManifest.xml

Make sure **both activities** are declared inside <application> tag — like this

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.intentsdemo">

    <application
        android:allowBackup="true"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/Theme.IntentsDemo">

        <!-- SecondActivity must be registered here -->
        <activity android:name=".SecondActivity" />

        <!-- MainActivity is the launcher screen -->
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>

    </application>

</manifest>
```

Important:

If you forget to declare SecondActivity, app will **crash immediately** when you try to open it.

Kotlin Files (in /java/com/example/intentsdemo/)

You should have exactly these **two files**:

MainActivity.kt

SecondActivity.kt

MainActivity.kt

```
package com.example.intentsdemo

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val etName = findViewById<EditText>(R.id.etName)
        val btnSend = findViewById<Button>(R.id.btnSend)

        btnSend.setOnClickListener {
            val name = etName.text.toString()
            val intent = Intent(this, SecondActivity::class.java)
            intent.putExtra("username", name)
            startActivity(intent)
        }
    }
}
```

SecondActivity.kt

```
package com.example.intentsdemo

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

class SecondActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_second)

        val tvResult = findViewById<TextView>(R.id.tvResult)
        val name = intent.getStringExtra("username")
        tvResult.text = "Welcome, $name!"
    }
}
```

Layout Files (in /res/layout/)

There must be **two layout XML files** with these exact names:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="20dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:id="@+id/etName"
        android:hint="Enter your name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

    <Button
        android:id="@+id/btnSend"
        android:text="Go to Next Screen"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
```

activity_second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="20dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/tvResult"
        android:textSize="22sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Hello!" />

</LinearLayout>
```

Gradle Settings (Auto-managed)

Usually you don't need to change anything here, but just to be safe:

In app/build.gradle check that it has:

```
plugins {  
    id 'com.android.application'  
    id 'org.jetbrains.kotlin.android'  
}  
  
android {  
    namespace 'com.example.intentsdemo'  
    compileSdk 34  
  
    defaultConfig {  
        applicationId "com.example.intentsdemo"  
        minSdk 24  
        targetSdk 34  
        versionCode 1  
        versionName "1.0"  
    }  
}
```

No changes required beyond this.

Now Run the App

1. Click **Build > Clean Project**
2. Then **Build > Rebuild Project**
3. Then **Run** 

If everything above is correct, your app will:

- Open on MainActivity
- Ask for name input
- On clicking the button → open SecondActivity
- Display “Welcome, [Name]!”

Folder structure

Correct folder structure for your **2-screen Intent App** (com.example.intentsdemo).
Android Studio (use **“Project”** view, not **“Android”** view).

Project Folder Structure

```
IntentsDemo/          ← Your project folder
|
├── app/
|   ├── src/
|   |   ├── main/
|   |   |   ├── AndroidManifest.xml  ← Both Activities declared here
|   |   |   |
|   |   |   ├── java/
|   |   |   |   └── com/
|   |   |   |       └── example/
|   |   |   |           └── intentsdemo/
|   |   |   |               ├── MainActivity.kt
|   |   |   |               └── SecondActivity.kt
|   |   |
|   |   └── res/
|   |       ├── layout/
|   |       |   ├── activity_main.xml
|   |       |   └── activity_second.xml
|   |       ├── values/
|   |       |   ├── strings.xml
|   |       |   └── mipmap-anydpi-v26/
|   |       |       └── ic_launcher.xml
|   |
|   |   └── kotlin/
|   |
|   └── test/
|
├── build.gradle (Module: app)
├── proguard-rules.pro
|
├── build.gradle (Project level)
└── settings.gradle
```

Common Mistakes That Cause “Keeps Stopping” Error

Check each of these carefully:

Problem	Fix
You forgot to declare SecondActivity in AndroidManifest.xml	Add <activity android:name=".SecondActivity" />
Layout file names don't match (typo like activity_second.xmlx)	Check under res/layout/ folder

setContentView(R.layout.activity_second) refers to a non-existent layout	Make sure file names match exactly
Wrong package name in AndroidManifest.xml	Should match Kotlin package (com.example.intentsdemo)
You didn't rebuild the project after adding new files	Run Build > Clean Project , then Rebuild Project
App not installed correctly (old version conflict)	Uninstall old version, then run again

How to Check Package and Layout Link

1. In **MainActivity.kt** – line at the top should match manifest:
2. package com.example.intentsdemo
3. In **AndroidManifest.xml** – top line:
4. package="com.example.intentsdemo"
5. In **MainActivity**:
6. setContentView(R.layout.activity_main)

Ensure activity_main.xml exists inside res/layout/.