**Intent: implicit & Explicit**

**Intent** is a messaging object that is used to communicate between components (**such as activities, services, or broadcast receivers**). It allows you to request actions from another component or app, facilitating navigation and interactions within or outside your app. Intents can be used to start an activity, service, or deliver a broadcast.

**1. Explicit Intent: Navigating Between Two Activities**

An **Explicit Intent** is used to navigate from one activity to another within the same application and can optionally pass data between them.

**Step-by-Step Guide:**

1. Create a new Android project in Android Studio.

2. Add a second activity:

- In Android Studio, go to `**File > New > Activity > Empty Activity**` and name it `**SecondActivity**`.

3. **Add buttons and text views** in the `**activity\_main.xml**` and `**activity\_second.xml**` layout files to make it user-interactive.

4. **Write the code** to start `**SecondActivity**` and pass data using the intent.

**Explicit Intent Example Code**

**MainActivity.kt (First Activity)**

```kotlin

// MainActivity.kt

} package com.example.explicitintentdemo

import android.content.Intent

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.widget.Button

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

// Button to navigate to SecondActivity using Explicit Intent

val btnGoToSecondActivity = findViewById<Button>(R.id.btnGoToSecondActivity)

btnGoToSecondActivity.setOnClickListener {

// Creating an explicit intent to navigate to SecondActivity

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

// Adding data to pass to SecondActivity

intent.putExtra("message", "Hello from MainActivity!")

// Starting SecondActivity

startActivity(intent)

}

}

**Line 1: val intent = Intent(this, SecondActivity::class.java)**

* **Intent**: An Intent is an object that represents a request to start another activity. It can carry data and be used to launch components like activities, services, or broadcast receivers.
* **this**: Refers to the current activity (MainActivity). It is used as the context from which the Intent is initiated.
* **SecondActivity::class.java**: This specifies the target component (in this case, another activity) that we want to launch. The ::class.java refers to the Java class corresponding to the SecondActivity in Kotlin.
* **Purpose**: This line creates an **explicit intent** to start SecondActivity. An explicit intent explicitly states which activity to open, unlike an implicit intent, which may use actions or categories to determine the target component.

**Line 2: intent.putExtra("message", "Hello from MainActivity")**

* **putExtra(String, String)**: This method attaches extra data to the intent. In this case, we are sending a key-value pair to the new activity:
  + "message": This is the key, a string identifier used to access the data in the next activity.
  + "Hello from MainActivity": This is the actual value that you are sending. It can be any type of data (e.g., String, Int, Boolean).
* **Purpose**: This line adds additional information ("Hello from MainActivity") that will be passed to SecondActivity. The target activity can retrieve this data using the key ("message").

**Line 3: startActivity(intent)**

* **startActivity()**: This method is called to actually start the new activity. It tells Android to process the intent and launch the specified target activity (SecondActivity).
* **Purpose**: This line starts SecondActivity by passing the previously created Intent object that contains both the target activity and the data to be sent.

**Summary:**

* **Intent creation (Intent(this, SecondActivity::class.java))**: You are preparing to launch SecondActivity from MainActivity.
* **Data attachment (putExtra("message", "Hello from MainActivity"))**: You are attaching a message to the intent, which will be passed along to SecondActivity.
* **Launching activity (startActivity(intent))**: Finally, this method triggers the transition from MainActivity to SecondActivity, sending the intent and any attached data.

**SecondActivity.kt (Second Activity)**

```kotlin

// SecondActivity.kt

package com.example.explicitintentdemo

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)

// Receiving the data from MainActivity

val message = intent.getStringExtra("message")

val textViewMessage = findViewById<TextView>(R.id.textViewMessage)

textViewMessage.text = message

}

}

**Line 1: val message = intent.getStringExtra("message")**

* **val message**: This declares a variable named message to hold the string data retrieved from the intent.
* **intent**: The intent object in an activity contains all the information that was passed to it when it was started. In this case, it's the Intent that was created in MainActivity.
* **getStringExtra("message")**: This method is used to retrieve data from the Intent. It fetches the value associated with the key "message", which was passed via putExtra in MainActivity:
*  If the "message" key exists in the intent, the corresponding string value will be stored in the message variable.
*  If the key doesn’t exist or the value is not a string, null will be returned.

**Line 2: val textViewMessage = findViewById<TextView>(R.id.textViewMessage)**

* **findViewById<TextView>(R.id.textViewMessage)**: This method is used to get a reference to a UI element from the XML layout. It locates the TextView component by its ID, R.id.textViewMessage.

**Line 3: textViewMessage.text = message**

* **textViewMessage.text**: This is how you set the text to be displayed inside the TextView.
* **message**: The content of the variable message (retrieved from the Intent) is assigned to the text property of the TextView. This means the string "Hello from MainActivity" (or any message passed) will now be displayed in the TextView with ID textViewMessage.

**Activity\_main.xml (Layout for MainActivity)**

```xml

<!-- activity\_main.xml -->

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<!-- Button to navigate to SecondActivity -->

<Button

android:id="@+id/btnGoToSecondActivity"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Go to Second Activity" />

</LinearLayout>

```

**activity\_second.xml (Layout for SecondActivity)**

```xml

<!-- activity\_second.xml -->

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:gravity="center">

<!-- TextView to display the passed message -->

<TextView

android:id="@+id/textViewMessage"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Message will appear here"

android:textSize="18sp"

android:textStyle="bold"/>

</LinearLayout>

```

**Declare SecondActivity in AndroidManifest.xml**

Don't forget to declare `SecondActivity` in your `AndroidManifest.xml`:

```xml

<activity android:name=".SecondActivity" />

```

**Steps Recap:**

1. Create a button in `MainActivity` that starts `SecondActivity`.

2. Pass data (a message) from `MainActivity` to `SecondActivity` using `Intent.putExtra()`.

3. Retrieve the message in `SecondActivity` using `intent.getStringExtra()`.

4. Display the message in a `TextView` in `SecondActivity`.

---

**2. Implicit Intent: Opening a Web Page**

An **Implicit Intent** is used to perform an action that is not explicitly specified within your app, such as opening a **URL in a web browser**, **sending an email**, or **dialing a phone number**.

**Step-by-Step Guide:**

1. Create a new Android project in Android Studio.

2. Add a button in the `activity\_main.xml` file that will trigger the implicit intent to open a website in a browser.

3. Write the code to trigger the intent to open a specific URL.

**Implicit Intent Example Code**

**MainActivity.kt**

```kotlin

// MainActivity.kt

package com.example.implicitintentdemo

import android.content.Intent

import android.net.Uri

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.widget.Button

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

// Button to open a web page using an implicit intent

val btnOpenWebsite = findViewById<Button>(R.id.btnOpenWebsite)

btnOpenWebsite.setOnClickListener {

// Create an implicit intent to open the website

val url = "https://www.google.com"

val intent = Intent(Intent.ACTION\_VIEW, Uri.parse(url))

// Start the intent

startActivity(intent)

}

}

}

```

**activity\_main.xml**

```xml

<!-- activity\_main.xml -->

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:gravity="center"

android:padding="16dp">

<!-- Button to open a website -->

<Button

android:id="@+id/btnOpenWebsite"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Open Google" />

</LinearLayout>

```

**Steps Recap:**

1. Create a button in `MainActivity`.

2. Set an `OnClickListener` for the button to trigger the **implicit intent**.

3. Create an **implicit intent** using `Intent.ACTION\_VIEW` with the URL you want to open.

4. Call `startActivity(intent)` to open the web browser with the specified URL.

**Real-Life Use Cases:**

**- Explicit Intent:** In a shopping app, you click on an item, and the app navigates to a detailed view of the item (second activity).

**- Implicit Intent:** Clicking a "Visit Store" button opens the official website of the store in a web browser.

**Summary**

- **Explicit Intent** is used to navigate between activities in your app and can pass data.

**- Implicit Intent** is used to perform actions using other apps on the device, like opening a browser, making a call, or sending a message.