

Implicit Intents

Implicit Intents do not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it.

Example 01

Create an application to start another activity by using implicit intents.

- Create a new project.
- Insert one button to the first activity.
- When the user clicks the button he should be directed to Gmail.
- You can use implicit intents for this.

```
public class MainActivity extends AppCompatActivity {  
    Button click;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        click= (Button) findViewById(R.id.btnClick);  
        click.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                Intent i = new Intent(Intent.ACTION_VIEW, (Uri.parse("http://www.gmail.com")));  
                startActivity(i);  
            }  
        });  
    }  
}
```

- First, a new intent object is created.
- Instead of specifying the class name of the intent, however, the code simply indicates the nature of the intent (to display something to the user) using the ACTION_VIEW option.
- The intent object also includes a URI containing the URL to be displayed.
- This indicates to the Android intent resolution system that the activity is requesting that a web page be displayed.
- The intent is then issued via a call to the startActivity() method.

Returning results from activities

Example 02

- Create a new android project.
- Create two activities.
- Add a TextView and a Button to the Main Activity.
- Add an EditText and a Button to the Second activity.
- In here, when the user clicks the button in the first activity he should be directed to the second activity.
- And in the second activity user should be able to enter a text to the EditText and when he click the button he should be directed to the first activity and at the same time the text entered should be displayed in the TextView.

Main Activity

- Lets start with the main activity.

```
public class MainActivity extends AppCompatActivity {

    TextView textView1;
    Button button1;
    static final int MY_REQUEST_CODE=2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView1= (TextView) findViewById(R.id.text);
        button1= (Button) findViewById(R.id.button);
        button1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent( packageContext: MainActivity.this,Second.class);
                startActivityForResult(intent,MY_REQUEST_CODE);
            }
        });
    }
}
```

- In here first we have to set a static final int MY_REQUEST_CODE=2 (you can give any integer number for this). It is an integer argument which is used to identify your request uniquely.
- Then you have to create an Intent object.
- After that you have to call the **startActivityForResults()** method. In here you have to pass two parameters to the method.
 1. The intent object that you have created.
 2. The request code that you have defined.

Second Activity

1. Then go to the second activity.
2. There you have one EditText and a Button.
3. Use the following lines of code for the second activity.

```
public class Second extends AppCompatActivity {
    EditText editText1;
    Button button1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        editText1=(EditText) findViewById(R.id.editText1);
        button1=(Button) findViewById(R.id.button1);
        button1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String message=editText1.getText().toString();
                Intent intent=new Intent();
                intent.putExtra( name: "MESSAGE",message);
                setResult(MY_REQUEST_CODE,intent);
                finish();
            }
        });
    }
}
```

4. In the **onClick** method of the button:
 1. First you have to create a String reference and you have to assign the text which was entered by the user.
 2. Then you have to create a new Intent object.
 3. Then by using **PutExtra** method you have to pass the text which was entered by the user to the intent object.
 4. After that by using **setResult** method you have to pass the **Request Code** that you have defined in the first activity and the intent reference that you have created in the second activity.

Main Activity

- In the main activity you have to call the **onActivityResult()** method. That method contains three parameters.
 1. The request code you passed to the **startActivityForResult()** method.
 2. The result code.
 3. The intent that carries the result data.

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    // check if the request code is same as what is passed here it is ;L
    if(requestCode==MY_REQUEST_CODE)
    {
        String message=data.getStringExtra (name: "MESSAGE");
        textView1.setText(message);
    }
}

```

- Then you have to check whether the request code returned for **onActivityResult()** is equal to the request code that you have assigned at the beginning.
- If it matches, then you have to set the text to the TextView in the main activity using **setText** method.

Linkify

- Linkify is a helper class that creates hyperlinks within Text View (and Text View-derived) classes through RegEx pattern matching.
- It converts the text and regular expression to the clickable links on the basis of the pattern match of text value and the regex.
- The Linkify class creates the links for web URL, email address, phone number, and map address by using pattern.

Example 03

- Create a new project.
- Add a TextView and set an id for it.
- Add the following code to the MainActivity.java file.

```

public class MainActivity extends AppCompatActivity {
    TextView link;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        link= (TextView) findViewById(R.id.link);
        link.setText("Android.com is the official website for Android");
        Linkify.addLinks(link, Linkify.WEB_URLS);
    }
}

```

- In here we have to first set a text for the TextView as the above figure. Since this is a Web Url the text should contain a valid web address (Eg: In here Android.com).
- Then by using Linkify.addLink method you can pass the **text view** and the **LinkifyMask**.
- As well as make sure to add the following code to the manifest file.

```
<uses-permission android:name="android.permission.INTERNET"/>
```

- In here since we are linking a Web address, we have to use the LinkifyMask as WEB_URLS.

Exercise 01 – Implicit Intents

Create an application that has an EditText and a Button. When the user enters an URL to the EditText and clicks the Button he should be directed to the entered URL.

Exercise 02 – Returning results from activities

Create the following Shopping Cart application. In the application there should be two activities as follows.

Main Activity

- There should be one TextView and a Button.
- When the user clicks the button, he should be directed to the second activity.

Second Activity

- There should be five products which are displayed on TextViews and in front of each text view there should be EditTexts.
- Add a Button and name it as **Calculate**.
- User should be able to type prices of the products on the EditTexts.
- When the user has finished entering the prices of the products the total should be calculated and when the user clicks the **Calculate Button** the total should be calculated, and the total should be transferred to the **Main Activity**.
- And it should be displayed on the TextView in the main Main Activity.

Exercise 03 – Returning results from activities

Create the following application for a Scientific Laboratory.

Main Activity

- There should be a TextView and a Button.
- When the user clicks the Button, he should be directed to the second activity.

Second Activity

- There should be TextViews which display Names and Color codes of the gas cylinders which are available in the laboratory. (You can define color codes for the cylinders as you wish) and (You can add up-to 5 or 6)
- There should be an Edit Text to enter the color code and when the user enters the color code and clicks the Button he should be directed to the main activity.
- And at the same time cylinder which matches to the color code should be displayed on the Text View of the Main Activity.
- You can use Switch cases to match the cylinders with color codes.

Exercise 04 –Linkify

Create an application that has three TextViews. The first one should display your name, your email address and your phone number. When the user clicks the email address, he should be directed to the default email client of the mobile. And when the user clicks the phone number, he should be directed to the default dialer of the mobile.