

For updated version, please click on  
<http://ocw.ump.edu.my>

# BCS3283-Mobile Application Development

## Chapter 7 Intent

Editor

**Dr. Mohammed Falah Mohammed**

**Faculty of Computer Systems & Software Engineering**  
**falah@ump.edu.my**

# Intent

- Aims
  - To learn how to passing and receive data between different activities.
- Expected Outcomes
  - Ability to handle explicit Intent
  - Ability to handle implicit Intent
  - Passing data to other activity
  - Returning data from launched activity
- References
  - <http://codetheory.in/>
  - <http://www.techotopia.com/>



# Introduction

- Most android application comprises one or more activities.
- In order to launch another like B Activity from A Activity, programmer have to use Intent.
- Intent is a way (like a message) to communicate between Android components and requesting an action.
- Incase there are multiple apps are capable of responding to the Intent, then Android enable the user to made a choice from a list of those apps.

# Intents Capability

Programmer will be able to accomplish a lot of things using intents like:

1. The ability of navigating from one activity to another
2. The ability to start a third party application (external app's activity) like using a camera, contact, pinning location, sending email and etc..

## Two types of intents are available:

- **Explicit Intent:** Launching an activity and passing information from one to another in the same application.
- **Implicit Intent:** In this case programmer can avoid coding his own activity and let Android automatically handle the activity by firing up the component from same/other applications that can do the job on our app's behalf.

## Example:

```
// Currently in MainActivity  
Intent intent = new Intent(MainActivity.this, ActivityB.class);  
startActivity(intent);
```

- Launching the B Activity.java from the MainActivity.java., for that purpose we use the Intent class and passing the current context (intent) by calling startActivity(). This will simply launch the B Activity from the same application right over MainActivity.

# Passing Data to other Activity

- Add data to intent in key-value pairs then retrieve it by the receiving activity.

```
// Currently in MainActivity
Intent intent = new Intent(MainActivity.this, ActivityB.class);
intent.putExtra("name", "Ali");
intent.putExtra("age", 21);
startActivity(intent);
```

In this case, two pieces of data (String and integer) have been transferred by using putExtra.

# Receiving Data in Other Activity

- To receive the data in B Activity by using Bundle object.

```
// Currently in ActivityB
Bundle extras = getIntent().getExtras();
if (extras != null) {
    String name = extras.getString("name");
    int age = extras.getInt("age");
}
```

Calling `getExtras()` on our Intent object enables us to call various methods like `getString()`, `getInt()`, `getChar()`, etc. but also we can use methods like `getStringExtra()` and `getIntExtra()` on the Intent object instead of working with bundles.



# Returning Data Launched Activity

- There is no way for transferred data to be returned from the B Activity to the first activity.
- However, launching the B Activity as a sub-activity of the first Activity can achieved returning the data.
- To do so, we have to start the first activity as a sub-activity by calling `startActivityForResult()` method instead of using `startActivity()`.
- This method enable us to identify the returned data from the sub-activity by passing a `REQUEST_CODE` value.

```
startActivityForResult( i , REQUEST_CODE);
```



# Returning Data from Launched Activity

- Returning data to the first activity:
  - Calling the finish() method in the sub-activity.  
finish() method creates a new intent object contains the returned data.
  - Calling the setResult() method of the enclosing activity to pass through it the return data.
- The result code is typically
  - **RESULT\_OK**: the result sent correctly
  - **RESULT\_CANCELED**: will be received in the parent activity if the event of sub-activity is crashed
  - May also be a custom value subject.

# Java Code Example

## Main Activity

Capturing information From the EditText, send it to the second activity, then do listening

```
public void MyActivity(View v){

    Intent intent=new Intent(MainActivity.this,My2ndActivity.class);
    intent.putExtra("ED1", ed1.getText().toString());
    startActivityForResult(intent,11);
    ed1.setText("");

}

protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if(requestCode==11 && resultCode==RESULT_OK){
        |   ed1.setText(data.getStringExtra("sms"));
    }
}

}
```



# Java Code Example

## Second Activity

Receiving the information and display it into TextView

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_my2nd);  
    et2=(EditText) findViewById(R.id.et2);  
  
    ((TextView) findViewById(R.id.tv2)).setText(getIntent().getStringExtra("ED1").toString());  
}
```



# Java Code Example

## Second Activity

Capturing information From the EditText and send it to the main activity (parent Activity)

```
public void send(View v) {  
    intent2=new Intent();  
    intent2.putExtra("sms",et2.getText().toString());  
    setResult(RESULT_OK,intent2);  
    finish();  
}
```

# Implicit Intent

- Implicit intent identify the activity to be launched by specifying the action to be performed and the type of data to be handled by the receiving activity.
- When execute the following implicit intent on an Android device, it will result in the designated web page appearing :

```
Intent i = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.ump.edu.my"));
```

- Android system will search for activities on the device that have registered the ability to handle ACTION\_VIEW requests.
- User will be prompted to choose from the available activity options incase of finding more than one match.