

Mobile applications

Master SDBIS/SIA

Octavian Dospinescu
2021

Android Intents

In everybody's language



Introduction

Intent is a type of object that allows us to claim/request the execution of an action from another component of the operating system. The **intents** are a means of communication between our application's components or between our application and other components of the operating system.

Intents are used to:

1. Start a new activity using: **startActivity ()** to whom we passed an intent which would contain the necessary information: name of the activity + additional information packed into an object of type **Bundle**.

Introduction

2. To start a service using: **startService ()** to whom we passed an intent which would contain the necessary information: name of the service + additional information wrapped in a Bundle object , like for an activity.

3. To send a Broadcast. A Broadcast is a message that is sent to all the applications that have previously set a listener: BroadcastReceiver. To send a Broadcast, we will use an Intent which will be passed to the method: **startBroadcast ()**.

Implicit vs Explicit

Do you remember when you were a child and you wanted something ?



Implicit intents

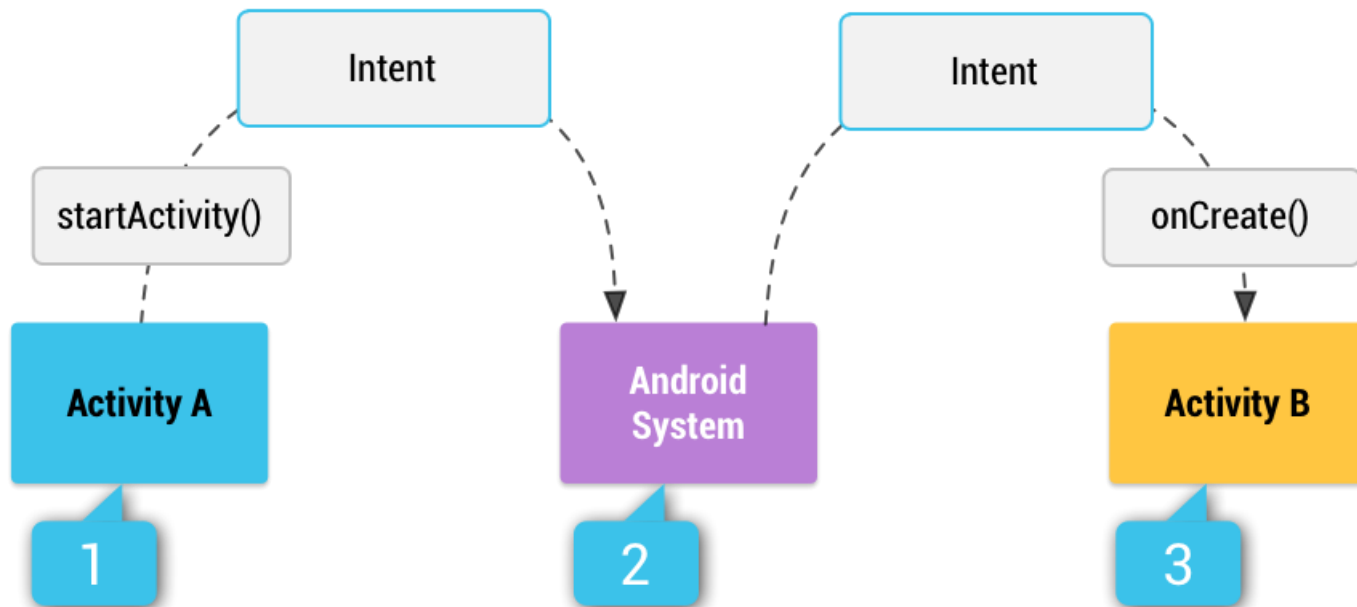
The pressure is on the "parent". If you ask for a pizza, parents gave the order and you only got the final product . You were not involved in the decision making process.

As happened in the case of default Intent: through them, we specify **what we want to do** and **the operating system will take care to find a "performer"**.

! Warning: there are some situations when the specified action can not be executed by any program. This case must be treated in our code ;)

According to:

<https://developer.android.com/guide/components/intents-filters>



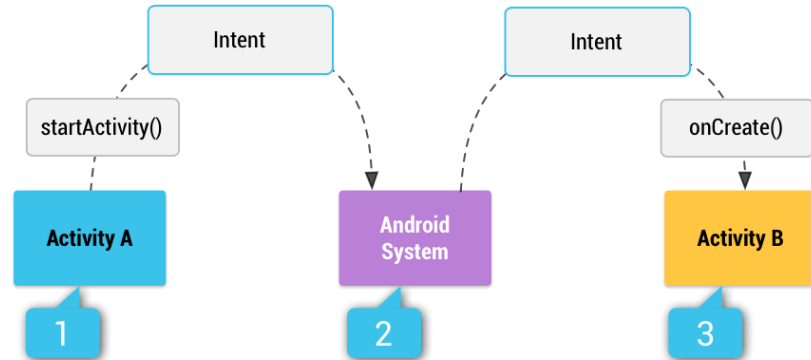
Implicit intents

How an **implicit intent** is delivered **through the system** to start another activity:

[1] Activity A creates an Intent with an action description and passes it to `startActivity()`.

[2] The Android System searches all apps for an intent filter that matches the intent.

When a match is found, [3] the system starts the matching activity (Activity B) by invoking its `onCreate()` method and passing it the Intent.



Explicit Intents

Fast forward a few years, do you remember when you've told parents that you want a phone? "Mom, I want a phone?"

And you left the rest in their care?

Or rather an **explicit** requirement (**order**) was : I want a Samsung Galaxy S5/6/7/8/9 ? 😊



Explicit Intents

The same is true for **explicit** Intents .

Through this type of intent, we tell the operating system **what we want to execute** . In this case we specify the class that is to be executed, usually to start a new activity within our application .



How to build an Intent

The main information that must be included in an Intent are :

- 1. The action to be executed** : we can specify a default action as listed in the Intent class , such as ACTION_VIEW, ACTION_SENT or we can specify a specific action of our application .
- 2. The name of the component** - this parameter is optional. If you specify the component name, the Intent will be **explicit**.
- 3. The type of data being transmitted** (and MIME type)
- 4. Category** - specifies the type of the component that should operate the intent
- 5. Additional information** - we can send information to the component that will open, as key-value using the method: **putExtra(key , value)**

Explicit intents: example 😊

```
// deschidem o noua activitate
btnSuperMan.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, SupermanActivity.class);
        intent.putExtra("nume", "Cei Jucaus");
        intent.putExtra("prenume", "Android");
        startActivity(intent);
    }
});
```

It opens **SuperManActivity** activity within our application when you press the button **btnSuperMan**.

We send 2 parameters to the activity, by using the Intent: **nume** and **prenume**.

Implicit intents – example 😊

```
// trimitem un mesaj folosind un Intent
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, "Intents are fun :)");
sendIntent.setType("text/plain");

// !!! Important: verificam daca actiunea pe care vrem s-o executam poate fi executata de macar o activitate
if (sendIntent.resolveActivity(getPackageManager()) != null) {
    startActivity(sendIntent);
}
```

We set the action to be executed and we send the “order” by: **startActivity()**.

The operating system will look for an activity that can execute our action and opens it, sending also the additional parameters. If there are several activities that can execute the action, a window will appear and we can choose a specific application.

Intent filters

Intents are used to perform a particular action, and the operating system will provide components that can execute our action. But how does the operating system do it?

By using **Intent Filters**.

Each component declares one or more Intent filters, telling (informing) the operating system what exactly it (the component) can do.

Intent filters

```
<activity
    android:name=".MainActivity"
    android:label="@string/app_name">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Intent filters are declared in [AndroidManifest.xml](#).

If we look in our application manifest, we already use an intent filter. It is defined in MainActivity activity and it is used to tell the operating system what activities to use when our application opens.

Intent filters – example 😊

We want to create a new application that can capture the intents of type `ACTION_SEND`. For this example we retrieve the text sent by the Intent and we will display it in a `TextView`.

So we will create 2 applications:

In the first application we create a button. Pressing it will send a default intent with action `ACTION_SEND` and a message using `putExtra`.

In the second application we will create an activity which defines an intent filter to capture intent of type `ACTION_SEND`.

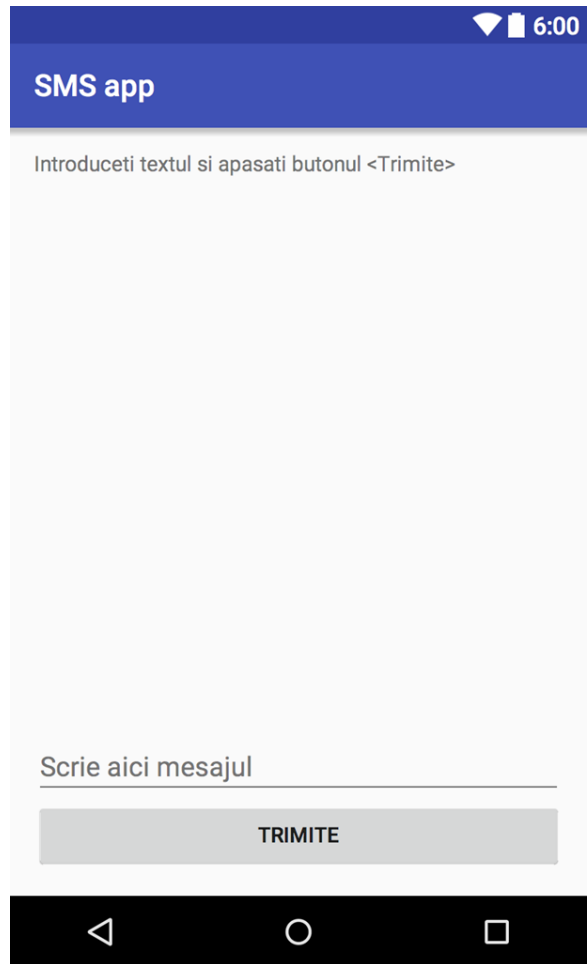
We will display the text sent by `putExtra()` in a `TextView`.

Example - Intent filters

In the **first** application we execute this code sequence inside the CLICK button's event. The button is TRIMITE.

```
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, mMessage.getText().toString());
sendIntent.setType("text/plain");
```

```
if (sendIntent.resolveActivity(getPackageManager()) != null) {
    startActivity(sendIntent);
}
```



Example - Intent filters

In the **second** application , we create an activity and we define an intent filter in AndroidManifest.xml. In the activity we will retrieve the data in Intent:

```
<activity android:name=".SendMessageActivity">
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="text/plain" />
    </intent-filter>
</activity>
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_send_message);

    tvMessage = (TextView) findViewById(R.id.message);

    Intent intent = getIntent();
    if(null != intent.getExtras())
    {
        tvMessage.setText(intent.getExtras().getString(Intent.EXTRA_TEXT));
    }
    else
        Log.i("TAVY","Inca niciun intent sosit");
}
```

SMS app

Introduceti textul si apasati butonul Trimite

Salutare AMA!

TRIMITE

SMS app

Introduceti textul si apasati butonul Trimite



Share with Exemple AMA

JUST ONCE

ALWAYS

Use a different app



Messaging



Bluetooth

Exemple AMA

Salutare AMA!

More details about **INTENTS** available at:

<https://developer.android.com/guide/components/intents-filters>

More details about **INTENTS TYPES** available at:

<https://developer.android.com/guide/components/intents-common>