

Föreläsning 4

- Intent
- Starta ny Activity
- BroadcastReceiver

Android - komponentbaserat

Android är komponentbaserat. Med det menas att en applikation i Android kan dra nytta av redan färdigskrivna komponenter.

Komponenter i Android är

- Activity, applikation
- Service, komponent som kör i bakgrunden
- Broadcast Receiver, mottagare av meddelanden från andra komponenter.
- Content Provider, gör data åtkomligt för andra komponenter

Activity

- En Activity du skriver kan t.ex. starta andra activities, både sådana som ingår i din applikation och sådana som finns installerade på Android-enheten.
- Och en applikation du skriver kan du göra tillgänglig att användas av andra applikationer.

Intent

Ett **Intent** är ett meddelande i Android. Intentet kan användas för att:

- Starta en Activity som finns på enheten
- Starta en Service som finns på enheten
- Skicka ett Broadcast Intent. Andra applikationer kan ta emot meddelandet.
- Ta emot ett Broadcast Intent. Broadcast Intentet kan vara skickat av en annan applikation eller av systemet

Intent – överföra data

Ett **Intent** kan innehålla data vilket kan användas av mottagaren.

Metoden *putExtra(key,value)* lägger till ett <key,value>-par i Intentet.

key ska vara en sträng och *value* kan vara:

- enkel variabeltyp (ex double)
- String
- arrayer med enkel variabeltyp (ex. double[])
- String-array (String[])
- Serializable (Object som implementerar Serializable)
- Parcelable (Object som implementerar Parcelable)
- Parcelable-array

Exempel:

```
Intent intent = new Intent(...);  
intent.putExtra( "name", "Ruth" );  
intent.putExtra ( "age", 26 );
```

Intent – överföra data

Mottagaren av ett **Intent** kan använda datan som lagrats i Intentet.

Metoden *getIntent()* ger referens till Intentet

Metoden *get...Extra(key)* / *get...Extra(key,default value)* returnerar ett värde. ... ersätts med typ av data.

Exempel:

```
Intent intent = getIntent();
```

```
String n = intent.getStringExtra( "name" ); // Om "name" ej finns så null
```

```
int age = intent.getIntExtra ( "age", -1 ); // Om "age" ej finns så -1
```

Starta en Activity

Det går utmärkt att starta en annan Activity från den Activity som exekverar. Skälet att man vill starta en Activity är att man har behov av en speciell tjänst. Denna tjänst kan exempelvis vara att skicka ett sms, visa en bild eller få en komplicerad beräkning utförd.

Man skapar ett Intent för att starta en Activity. Innehållet i Intentet avgör vilken Activity som startas.

- Man kan *explicit* (i klartext) uttrycka den Activity man önskar starta. Det handlar då om att starta en Activity i samma applikation. Nedanstående kod startar Activityn ActivityToStart.

```
Intent intent = new Intent( context, ActivityToStart.class );  
startActivity( intent );
```

- Man kan *implicit* uttrycka vad man vill ha för speciell tjänst av en Activity. Och om en sådan finns startas den. Nedanstående kod startar en App som hanterar telefonsamtal.

```
Intent intent =  
    new Intent(Intent.ACTION_DIAL,Uri.parse("tel:+46708123456"));  
startActivity( intent );
```

Starta Activity för returdata

Det går utmärkt att starta en activity för att erhålla någon form av resultat. Activityn ska startas med metoden *startActivityForResult*

```
public static int NAME = 12; // requestcode  
public static String ANSWER = "name";
```

```
Intent intent = new Intent( context, NameInput.class );  
startActivityForResult( intent, MainActivity.NAME );
```

För att erhålla resultatvärde från den startade activityn ska metoden *onActivityResult* överskuggas.

```
protected void onActivityResult(int requestCode, int resultCode, Intent  
data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    if(resultCode==Activity.RESULT_OK && requestCode==NAME) {  
        String str = data.getStringExtra(ANSWER);  
        :  
    }  
}
```

Activity som ger returdata

En activity, vilken startats med *startActivityForResult*, överför data och avslutar sig själv när uppgiften är färdig.

```
String str;
```

```
:
```

```
Intent result = new Intent();
```

```
result.putExtra(MainActivity.ANSWER, str);
```

```
setResult(Activity.RESULT_OK, result);
```

```
finish();
```


Explicit Intent

Ett explicit Intent används för att starta en Activity (eller annan komponent) i applikationen.

Konstruktor

```
public Intent (Context packageContext, Class<?> cls)
```

Exempel

```
Intent intent = new Intent( context, ActivityToStart.class );  
startActivity( intent );
```

En Activity som startas ska vara deklarerad application-taggen i manifestet.

```
<application  
:  
  <activity  
    android:name=".ActivityToStart"  
    android:label="@string/title_activity_to_start" >  
  </activity>  
:  
</application>
```

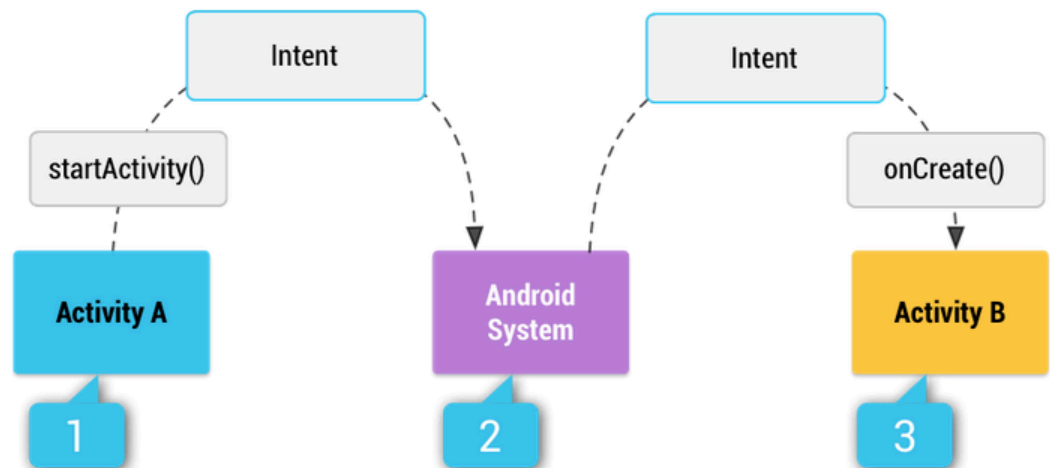
Implicit Intent

Ett implicit Intent används för att starta en Activity på Android-enheten. Med hjälp av Intentet beskriver man Activityn som ska startas.

- **Action** Ett stort antal olika actions-konstanter finns i Intent-klassen. Det går bra att deklarera egna actions.
- **Data** Vilken typ av data det handlar om och/eller en URI till datan
- **Category** Kompletterande information.

Om det finns matchande Activity så startas den. Om det finns flera så presenteras en lista för användaren att välja komponent från.

Figuren visa hur ett Intent levereras till en Activity som startas i systemet.



Intent filter

Ett Intent-filter används för att deklarera vilken typ av action/data/categori som en Activity kan hantera. Intent-filtret är i Manifest-filen och är en del av Activity-taggen.

Exempel på Intent filter

```
<activity android:name="ShareActivity">
  <!-- This activity handles "SEND" actions with text data -->
  <intent-filter>
    <action android:name="android.intent.action.SEND"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="text/plain"/>
  </intent-filter>
</activity>
```

Det går bra att lägga till fler action-taggar / category-taggar / data-taggar i filtret. En Activity som ska startas implicit måste alltid innehålla ovanstående category-tag

```
<category android:name="android.intent.category.DEFAULT"/>
```

Intent resolution

Att matcha ett Intent med ett Intent-filter går till så här:

1 Action

- Action i Intentet måste exakt matchas av en actiontagg i filtret.
- Ett filter utan actiontagg kan inte matchas
- Ett intent som inte innehåller någon action matchas av alla filter med någon actiontagg.

2 Category

- Varje category i Intentet måste matcha en category i filtret.
- Ett intent utan category matchas av alla filter

3 Data

- Se Kurslitteraturen s 186.

Speciella typer av Intent

Pending Intent

Ett Pending Intent är ett Intent som ska starta en annan Activity. Men detta ska ske någon gång i framtiden, t.ex. vid en Notification. I ett Pending Intent kapslas den avsändande Activityns permissions in. Och dessa måste räcka för att starta den nya Activityn.

Sticky Intent

Ett intent som finns lagrat i systemet och som man direkt kan få tillgång till. Det innebär att det inte behövs någon Broadcast Receiver för att ta del av dessa,

Ett par exempel på Sticky Intent är batteriets laddningsnivå och om enheten är uppkopplad via WiFi.

Broadcast Receiver

En Broadcast Receiver används framför allt för att ta emot meddelande från systemet om t.ex. batteriets laddning eller om enheten har ändrats till flygplansläge.

En lång lista över meddelanden som man kan lyssna efter finns i API-dokumentationen över Intent. ([d.android.com/...](http://d.android.com/))

Dessutom deklareras meddelandetyper även i andra klasser.

En Broadcast Receiver ska ärva `BroadcastReceiver` och implementera metoden *onReceive*.

```
public class PhoneReceiver extends BroadcastReceiver {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
    }  
}
```

`onReceive`-metoden ska avslutas snabbt. Om man ska göra någon tidskrävande så ska det ske asynkront. Mer om detta på föreläsning om trådar.

Broadcast Receiver

Här är ett exempel på en BroadcastReceiver som lyssnar efter inkommande telefonsamtal.

```
public class PhoneReceiver extends BroadcastReceiver {
    public void onReceive(Context context, Intent intent) {
        if (intent != null) {
            String state = intent.getStringExtra(TelephonyManager.EXTRA_STATE);
            Log.w("PhoneReceiver", "EXTRA_STATE="+state);
            if (state.equals(TelephonyManager.EXTRA_STATE_RINGING)) {
                String phoneNumber = intent
                    .getStringExtra(TelephonyManager.EXTRA_INCOMING_NUMBER);
                Log.w("PhoneReceiver", "EXTRA_INCOMING_NUMBER="+phoneNumber);
            }
        }
    }

    // Intent startIntent = new Intent(context, MainActivity.class);
    // Intent startIntent = new Intent("se.mah.tsroax.f4startactivity.MainActivity");

    // startIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    // context.startActivity(startIntent);
}
```

Sist i onReceive ges exempel på hur man kan starta en Activity från Broadcast Receivern. Avaktivera en av de översta och de två understa raderna. Activity-klassen i rad två måste anpassas till en Activity-klass som finns på enheten.

Broadcast Receiver

En Broadcast Receivers kan registreras i manifestet eller via kod.

Om den registreras via manifestet lyssnar den även då Activityn inte är startad. Men Activityn måste vara exekverad minst en gång.

Att lyssna efter inkommande samtal kräver ett permission:

android.permission.READ_PHONE_STATE

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="se.mah.tsroax.f4broadcastreceiver" >
```

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

```
  <application
```

```
    :
```

```
    <activity
```

```
      :
```

```
    </activity>
```

```
    <receiver android:name=".PhoneReceiver" >
```

```
      <intent-filter>
```

```
        <action android:name="android.intent.action.PHONE_STATE" />
```

```
      </intent-filter>
```

```
    </receiver>
```

```
  </application>
```

```
</manifest>
```


Broadcast Receiver

En Broadcast Receivers kan registreras i manifestet eller via kod.

Om den registreras via kod (i onResume) så ska den avregistreras (i onPause). Broadcast Receivern är endast aktiv då Activityn är aktiv.

```
public class MainActivity extends Activity {  
    private PhoneReceiver phoneReceiver = new PhoneReceiver(this);  
    private IntentFilter filter =  
        new IntentFilter(TelephonyManager.ACTION_PHONE_STATE_CHANGED);  
    :  
    @Override  
    protected void onResume() {  
        super.onResume();  
        registerReceiver(phoneReceiver, filter);  
    }  
  
    @Override  
    protected void onPause() {  
        super.onPause();  
        unregisterReceiver(phoneReceiver);  
    }  
}
```

```
private PhoneReceiver phoneReceiver = new PhoneReceiver(this);
```

Om metod i aktivitet ska anropas från BroadcastReceivern kan referens bifogas vid konstruktion.