

Intent

Intent

- Un'Intent è che un messaggio contenente un insieme di informazioni che vengono inviate ad un altro oggetto il quale, leggendole, effettua alcune operazioni stabilite dal programmatore.
- Le informazioni nell'Intent sono
 - Component name
 - specifica in modo esplicito il componente che lo riceverà
 - Action
 - è una stringa che identifica il tipo di operazione che sarà eseguita
 - Data
 - l'URI della risorsa che vogliamo elaborare
 - Category
 - è una stringa contenente informazione sul tipo di componente che lo può ricevere

Intent espliciti

Servono a lanciare Activity

ES1

```
Intent intent = new Intent(EsplicitIntentTestActivity.this, SecondActivity.class);  
startActivity(intent);
```

ES2

```
Intent intent = new Intent();  
ComponentName component = new ComponentName(EsplicitIntentTestActivity.this,  
                                              SecondActivity.class);  
  
intent.setComponent(component);  
startActivity(intent);
```

ES3

```
Intent intent = new Intent();  
Class<?> destination = Class.forName("it.apogeo.android.cap04.lifecycleactivitytest.LifecycleActivityTestActivity");  
ComponentName component = new ComponentName(EsplicitIntentTestActivity.this, destination);  
intent.setComponent(component);  
startActivity(intent);
```

Dichiarazione

- Manifest

```
<application
    android:icon="@drawable/icon"
    android:label="@string/app_name" >
    <activity
        android:name=".ExplicitIntentTestActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name="SecondActivity" >
    </activity>
</application>
```



I Task

- Ad un applicazione viene associato almeno un task
 - numero che identifica una sequenza di activity
- Le activity si alternano sul display
 - c'è uno stack di Activity
- Se non specificato le activity condividono lo stesso task
- Per lanciare una Activity in un nuovo task devono fare due cose:
 - dichiarare un attributo taskaffinity
 - `android:taskAffinity=":Other"`
 - settare nell'Intent un flag specifico
 - `Intent.FLAG_ACTIVITY_NEW_TASK`

Intent ed Action

- una action è una stringa che identifica il tipo di operazione che sarà eseguita
 - `Intent.ACTION_DIAL`
 - “android.intent.action.DIAL”
 - *it.apogeo.android.cap04.senderproject.intent.action.CUSTOM_ACTION*
- Sintassi della stringa
 - `<package>.intent.action.<action name>`

Action custom

Receiver

- Si usa un Intent filter per dichiarare la volontà di gestire l'azione nel receiver

```
<intent-filter>
    <action android:name="cap04.senderproject.intent.action.CUSTOM_ACTION">
    </action>
    <category android:name="android.intent.category.DEFAULT">
    </category>
</intent-filter>
```

Sender

- Si usa un Intent a cui si imposta l'action opportuna

```
Intent intent = new Intent();
intent.setAction(CUSTOM_ACTION);
startActivity(intent);
```

Action generiche

- Costanti della classe Intent

- ACTION_MAIN
- ACTION_VIEW
- ACTION_EDIT
- ACTION_DIAL
- ACTION_CALL
- ACTION_SEND
- ACTION_SEARCH

- Esempio di esecuzione della activity per inserire un numero

```
Intent intent = new Intent() ;  
intent.setAction(Intent.ACTION_DIAL) ;  
startActivity(intent) ;
```


Intent, action e dati

- In generale una coppia action/dato specifica l'activity da eseguire
- la forma del dato è una URI (RFC 2396)
 - scheme://host:port/path
- Esempi
 - ACTION_VIEW content://contacts/people/1
 - ACTION_DIAL content://contacts/people/1
 - ACTION_VIEW tel:123
 - ACTION_DIAL tel:123
 - ACTION_EDIT content://contacts/people/1
 - ACTION_VIEW content://contacts/people/

Filtri sul contenuto

- Si può impostare dei filtri generici sul mimeType
 - cioè il tipo di contenuto
 - immagini (image/jpeg)
 - video (audio/mpeg4-generic)
 - contatti (specifico di Android)
 - etc.
- Esempio di gestione contatti

```
<intent-filter>  
  <action android:name="android.intent.action.VIEW"></action>  
  <data android:mimeType="vnd.android.cursor.dir/contact"></data>  
  <category android:name="android.intent.category.DEFAULT"></category>  
</intent-filter>
```

Intent e categorie

- Le categorie classificano gli Intent ed Intent-filter
 - Quindi permettono di raggruppare logicamente le Activity

Es. aggiunta di una categoria nell'Intent

```
intent.addCategory(Intent.CATEGORY_LAUNCHER);
```

Es. definizione di una categoria nell'Intent Filter

```
<intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
</intent-filter>
```

Affinché una Activity sia compatibile con un Intent è necessario che tutte le categorie impostate nell'Intent stesso, siano tra quelle elencate nell'IntentFilter.

Package manager

- La classe che permette di usare le categorie si chiama Package Manager
 - Permette di scoprire quali app soddisfano un intent

Esempio di Intent filter

```
<intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
    <category android:name="android.intent.category.HOME" />  
    <category  
android:name="android.intent.category.DEFAULT"></category>  
</intent-filter>
```

Esempio di ricerca delle activity che possono essere eseguite per lanciare una app

```
Intent intent = new Intent(Intent.ACTION_MAIN);  
intent.addCategory(Intent.CATEGORY_LAUNCHER);  
PackageManager packageManager = getPackageManager();  
List<ResolveInfo> infoList =  
    packageManager.queryIntentActivities(intent, 0);
```

Activity che restituiscono valori

- Lanciare una activity per far ottenere un valore (URI) di ritorno

```
Intent intent = new Intent(Intent.ACTION_VIEW);  
intent.setData(android.provider.ContactsContract.Contacts.CONTENT_URI);  
startActivityForResult(intent, CHOOSE_CONTACT_CODE);
```

- Quando l'activity child “esce” viene chiamato il metodo

```
protected void onActivityResult(int requestCode, int resultCode, Intent  
data)
```

Esempio

```
protected void onActivityResult(int requestCode, int resultCode, Intent  
data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    if(requestCode==CHOOSE_CONTACT_CODE) {  
        if(resultCode == Activity.RESULT_OK) {  
        }else if(resultCode == Activity.RESULT_CANCELED) {  
        }else{}  
    }else{}  
}
```

Generazione del risultato

- Nell'activity child
 - Gestisco l'esito
 - Posso impostare il dato nell'intent che ritorno
- Dati di ritorno
 - URI del dato
 - Extra
 - `public Bundle getExtras ()`
 - `public Intent putExtras (Bundle extras)`

Esito positivo

```
setResult(Activity.RESULT_OK, dataIntent);  
finish();
```

Esito negativo

```
setResult(Activity.RESULT_CANCELED, dataIntent);  
finish();
```

oppure solo `finish()`