INTENTS

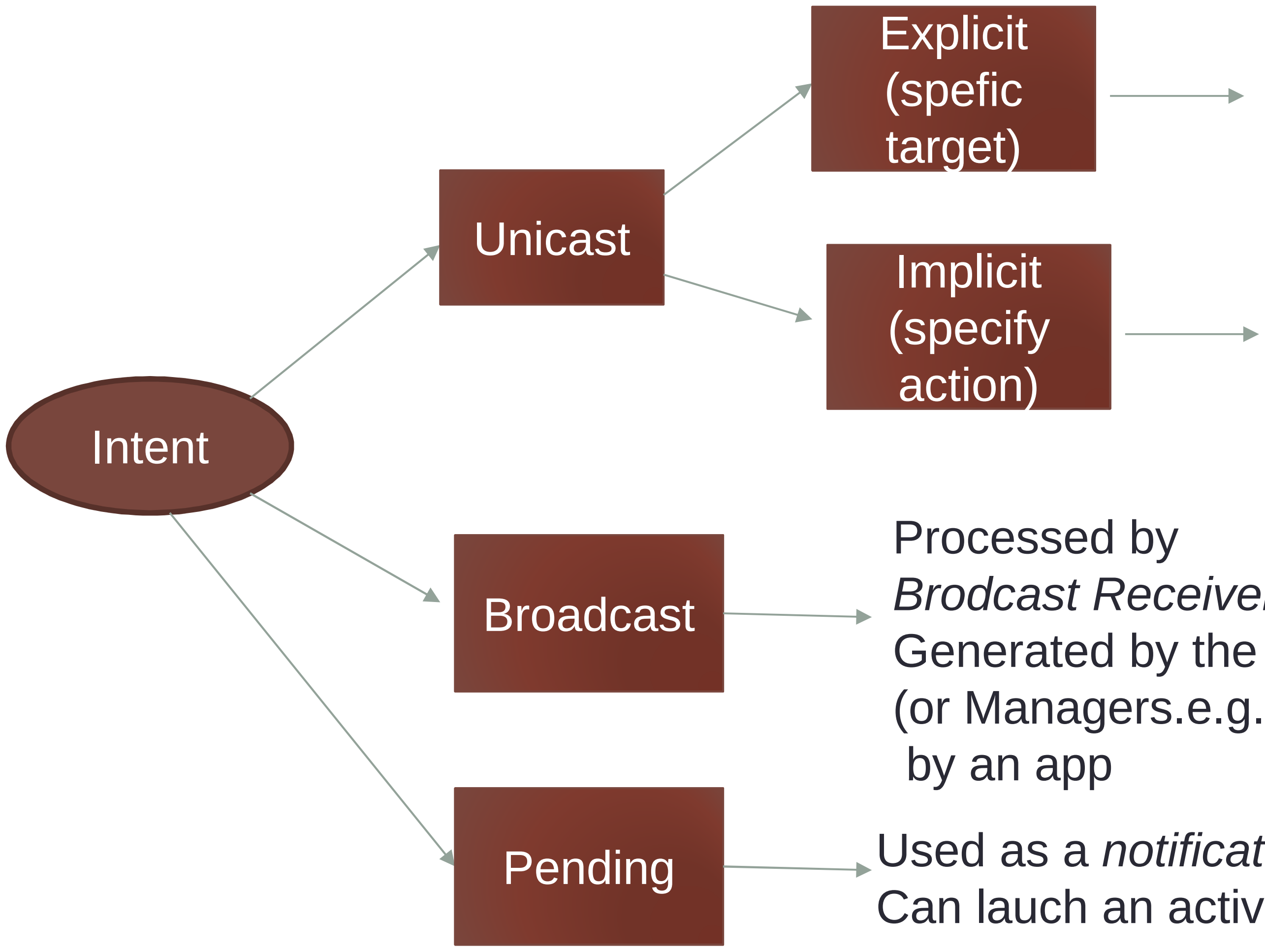
Intents (concept)

- Intents are like messages that activate software components (activities,servicies,broadcast receivers)
- An Intent may be **explicit** when it has exactly one target
- or **implicit** otherwise: it just specifies the *action* the component should provide.
- An intent can also be **pending**, meaning that some component will be activated in the future
- It can be **broadcast** when it announces something to **broadcast receivers**

Intents (concept)

- There are a lot of predefined actions (unicast or bcast)
- User can define new actions
- An app may be not allowed to generate some specific action as they are reserved to the system

Intents



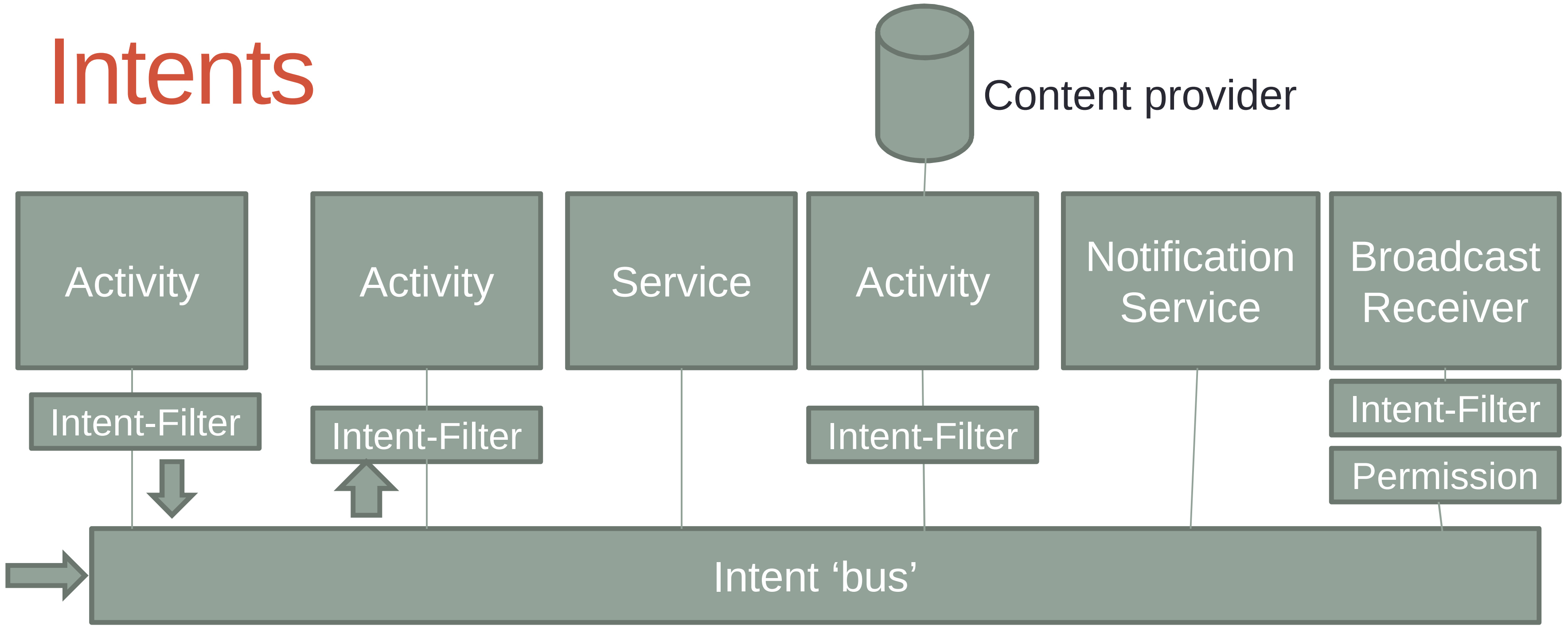
Start an Activity
Start a Service

No target found→ **error**
Only one target → **started**
Many targets→**user choice**

Processed by
Brodcast Receivers
Generated by the system
(or Managers.e.g. AudioManager) or
by an app

Used as a *notification*
Can lauch an activity (later)

Intents

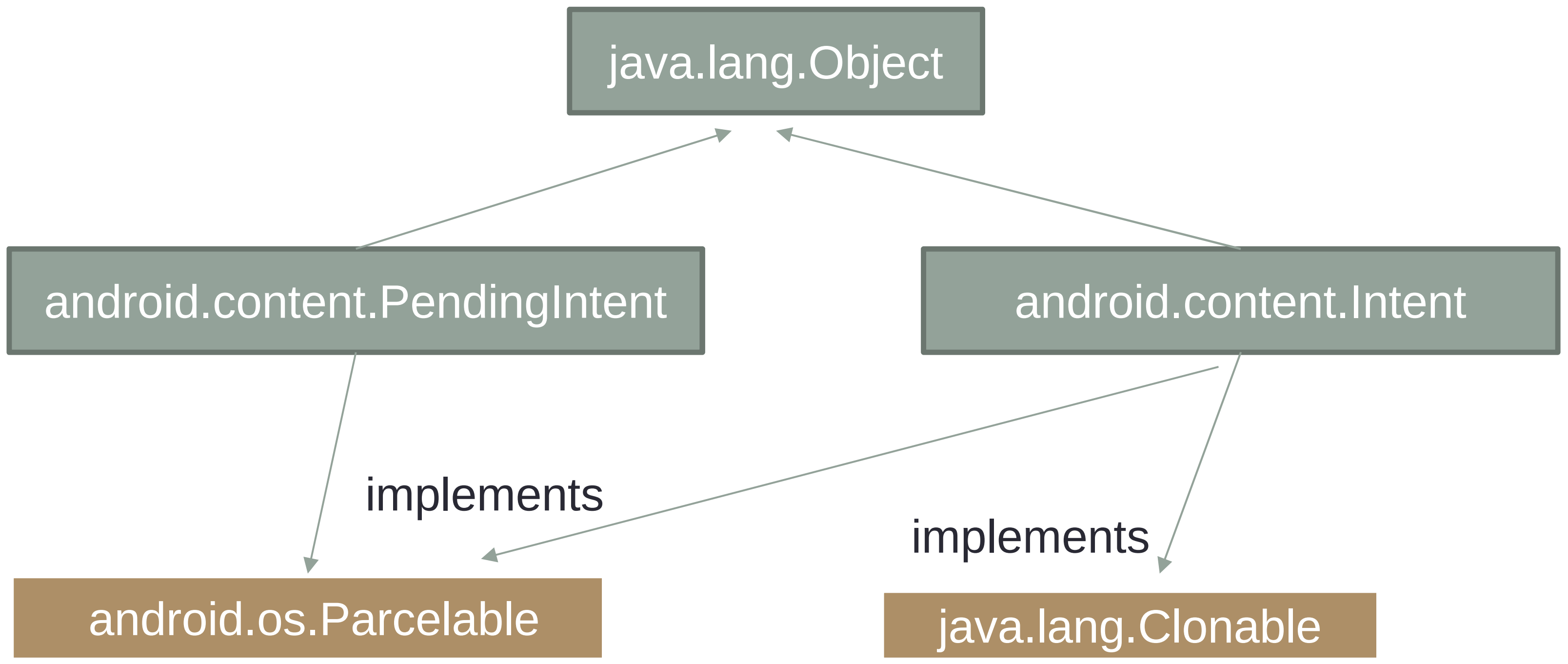


Examples:

1. Activity A launches Activity B
2. Activity A launches a Service (see future lectures)
3. An activity A wants to perform an action on some data (i.e., to see contacts)
4. An activity wants to notify something to the user (icon in the notification bar)
5. System notifies some event to 'all' (for example, TIME_TICK)

For security reason: It's better to start a service explicitly, i.e. not using filters
(see official documentation)

SW hierarchy



Intent Fields (java class)

- Action:
  - a string representing the operation (action) to perform
    - For example: «**android.intent.action.MAIN**» (it goes in the manifest file)
    - Referred symbolically in the code as **Intent.ACTION_MAIN**
- Category:
  - String representing additional information about the component that can manage the intent
    - For example: «**android.intent.category.LAUNCHER**»
    - Symbolically ad **Intent.CATEGORY_LAUNCHER**
- Data:
  - A URI that references data to operate on (*scheme://authority/path*)
    - For example: ***content://contacts/people/1*** → Display information about the person whose identifier is "1"

Intent Fields (java class)

- Component
  - The name of the component to start (used when the component that can handle the intent is known).
- Extras
  - Key-value pairs to carry additional information required to perform the requested action
    - For example, if the action it to send an e-mail message, one could also include extra pieces of data here to supply a subject, body, etc.
- Flags
  - Behaviours, e.g. don't push the activity onto the stack, print log, etc.

Explicit intent

- Activities are independent from each other and interact through **Intents**

- The explicit intent targets one specific activity, for example just to change the screen

- They are also used to start a *service* (future lecture)

Example

```java
public class MainActivity extends Activity
```

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

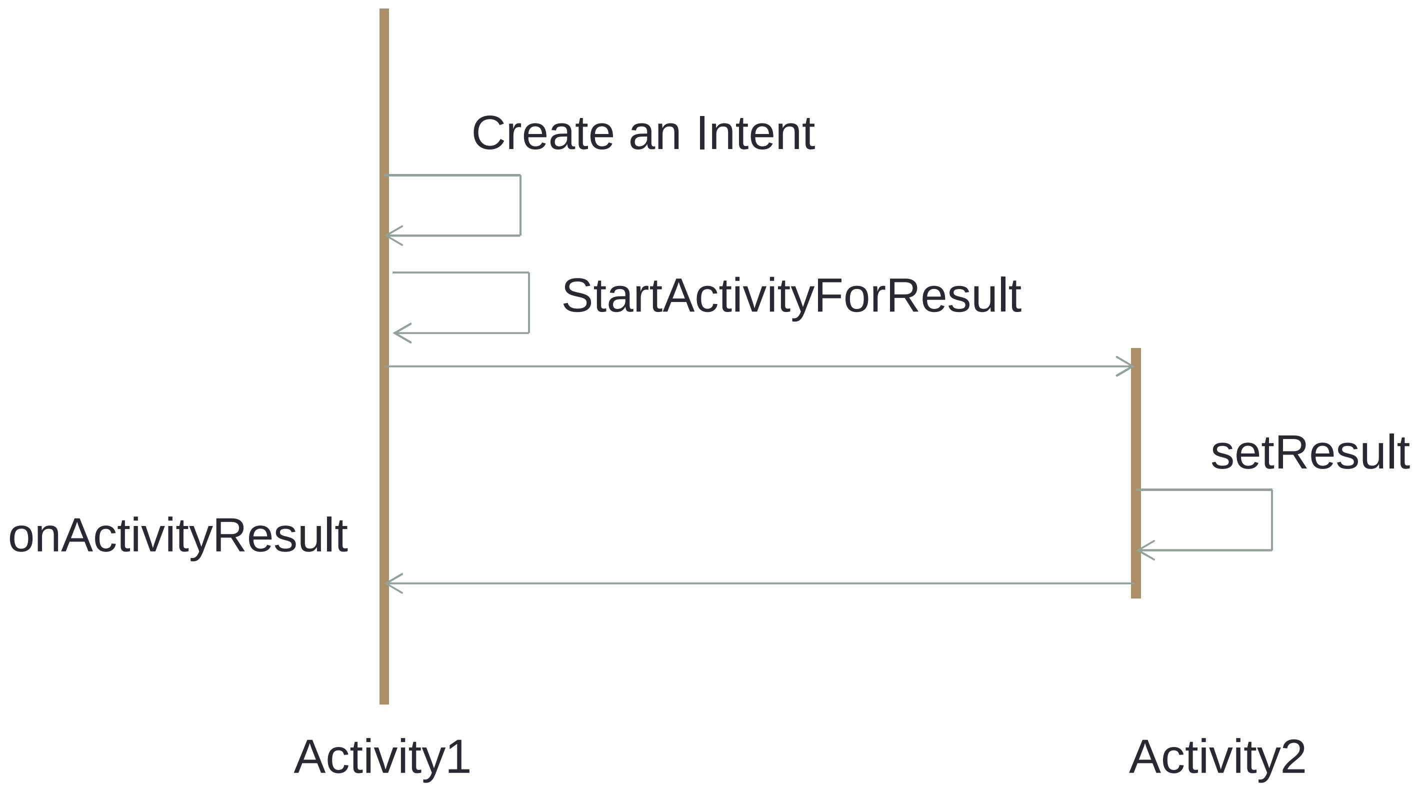
```xml
<application android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name=".MainActivity"
              android:label="@string/app_name">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER"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

```java
public class SecondActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
    }
```

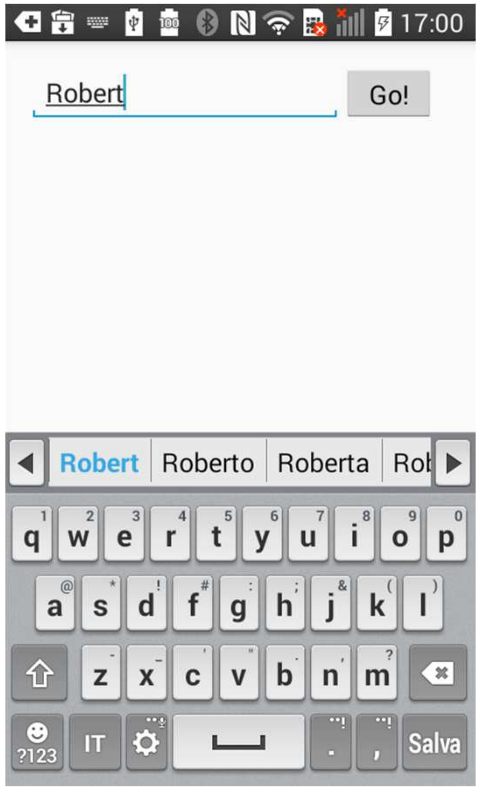
Starting an activity and getting results

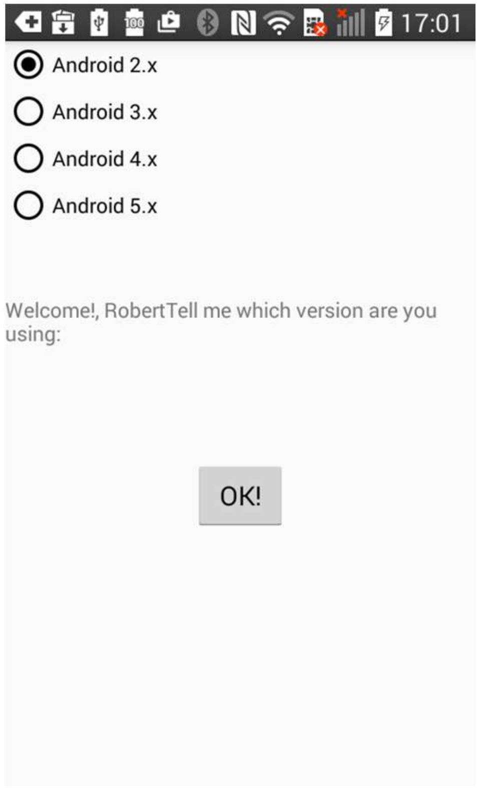
- Allows to call an activity and get results
- The calling Activity will not wait
- The called activity will issue **setResult** method call
- This causes the **onActivityResult** method of the calling activity to be executed

Starting an activity and getting results



ActivityCalls(demo)





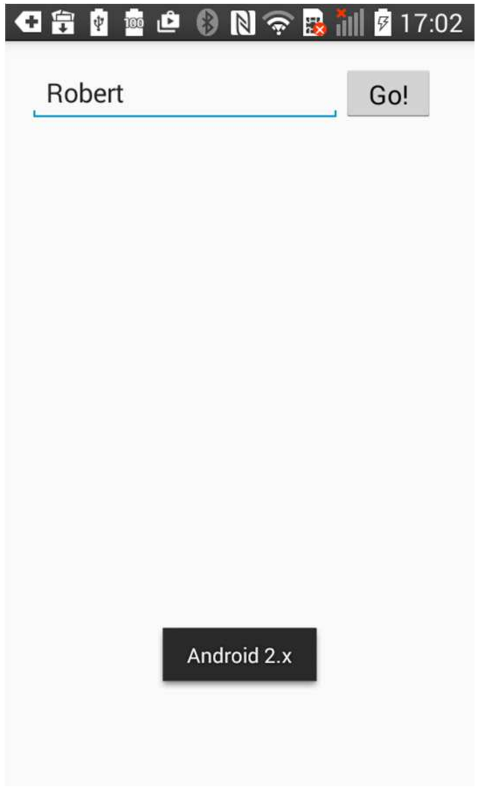
[✓ Android 2.x]

Android 3.x

Android 4.x

Android 5.x

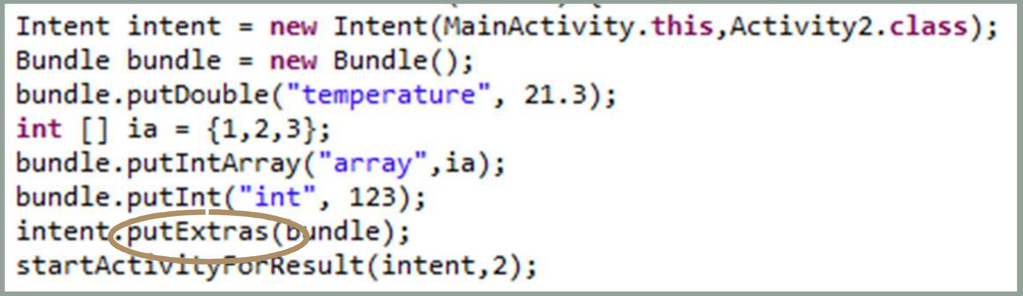
Welcome!, RobertTell me which version are you using:



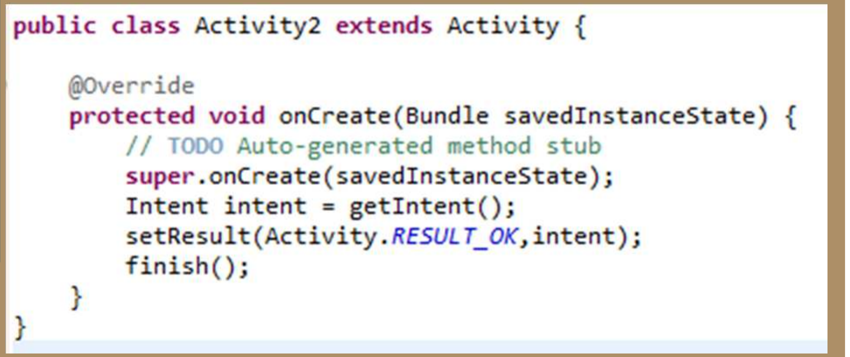
Passing data to the new activity

- When an Activity A has to pass some data to anohter activity it needs to set extra fields in the intent object

Passing data via a bundle

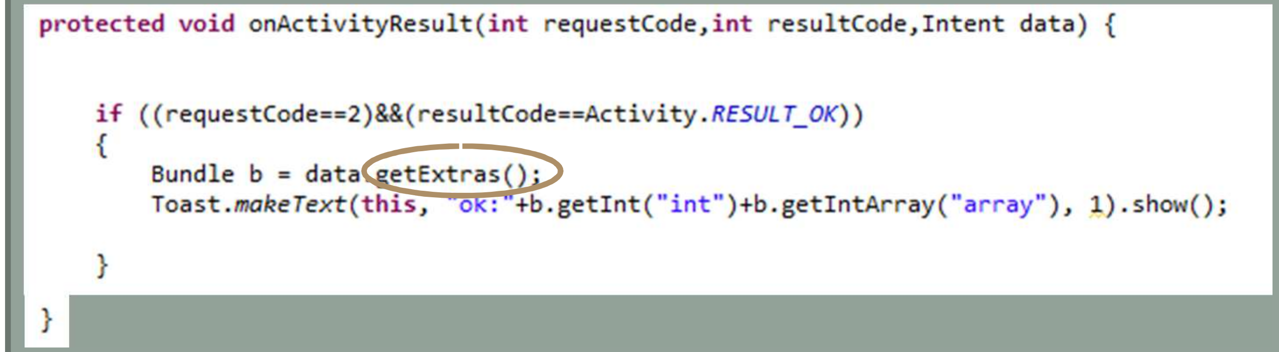


```
Intent intent = new Intent(MainActivity.this,Activity2.class);
Bundle bundle = new Bundle();
bundle.putDouble("temperature", 21.3);
int [] ia = {1,2,3};
bundle.putIntArray("array",ia);
bundle.putInt("int", 123);
intent.putExtras(bundle);
startActivityForResult(intent,2);
```



```
public class Activity2 extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        Intent intent = getIntent();
        setResult(Activity.RESULT_OK,intent);
        finish();
    }
}
```

Activity2



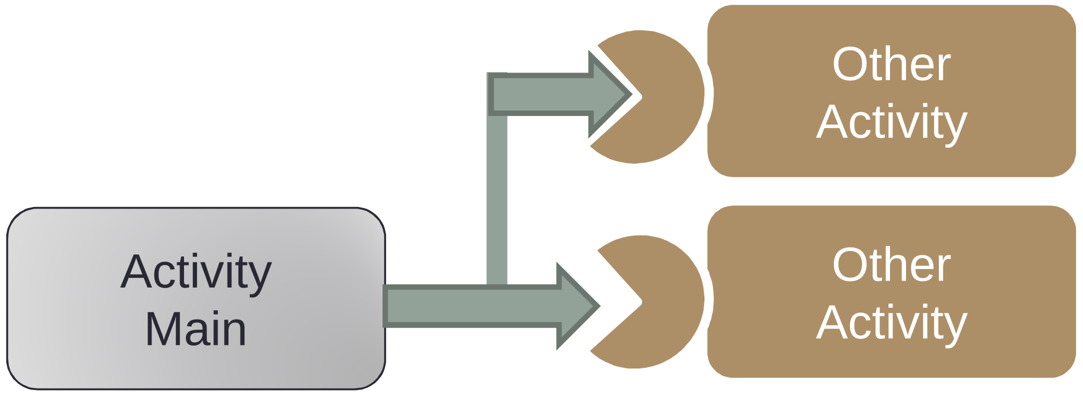
```
protected void onActivityResult(int requestCode,int resultCode,Intent data) {


    if ((requestCode==2)&&(resultCode==Activity.RESULT_OK))
    {
        Bundle b = data.getExtras();
        Toast.makeText(this, "ok:"+b.getInt("int")+b.getIntArray("array"), 1).show();

    }
}
```

Passing data via a bundle

- In the example, data are just echoed back to the caller
- The called activity gets the intent via getIntent method
- The called activity sets no screen and it is immediately finished

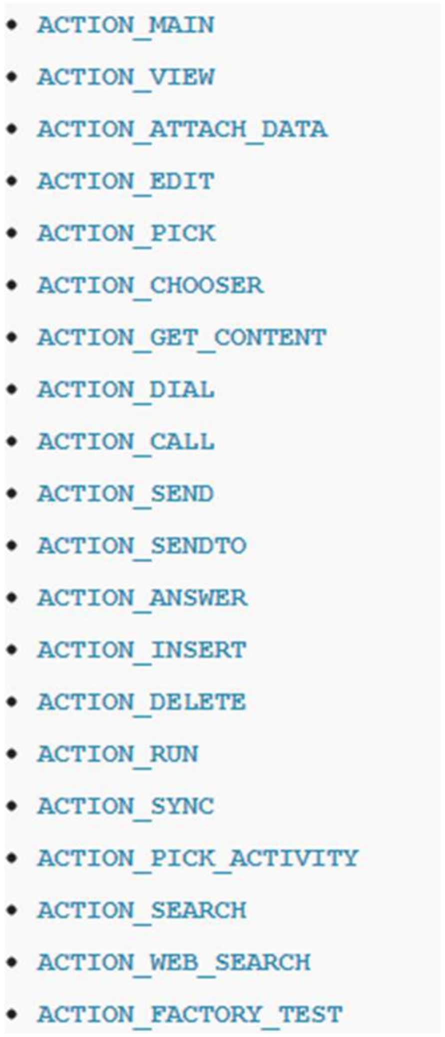
- We will see that for computations without UI, services or threads are more suitable

Implicit intents



- The Intent doesn’t specify the Activity to start, but only an “Action”
- Intents declares their ability to perform actions in the manifest file
- There are several predefined actions in the ‘system’ to choose from
- A user can define its own action as well

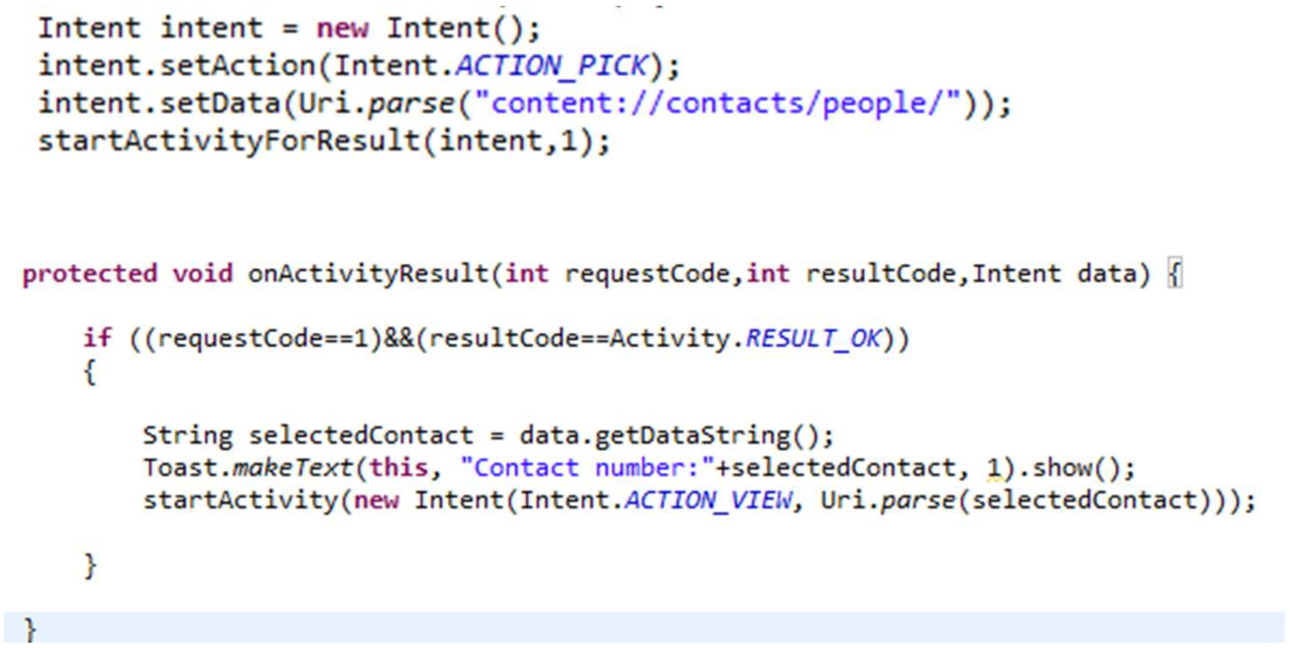
Example of system defined actions



- ACTION_MAIN
- ACTION_VIEW
- ACTION_ATTACH_DATA
- ACTION_EDIT
- ACTION_PICK
- ACTION_CHOOSER
- ACTION_GET_CONTENT
- ACTION_DIAL
- ACTION_CALL
- ACTION_SEND
- ACTION_SENDTO
- ACTION_ANSWER
- ACTION_INSERT
- ACTION_DELETE
- ACTION_RUN
- ACTION_SYNC
- ACTION_PICK_ACTIVITY
- ACTION_SEARCH
- ACTION_WEB_SEARCH
- ACTION_FACTORY_TEST

Example

- Select a contact from the contact list
- Show the contact ID on the screen and view the details



```
Intent intent = new Intent();
intent.setAction(Intent.ACTION_PICK);
intent.setData(Uri.parse("content://contacts/people/"));
startActivityForResult(intent,1);


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

    if ((requestCode==1)&&(resultCode==Activity.RESULT_OK))
    {

        String selectedContact = data.getDataString();
        Toast.makeText(this, "Contact number:"+selectedContact, 1).show();
        startActivity(new Intent(Intent.ACTION_VIEW, Uri.parse(selectedContact)));

    }

}
```

Example of action/data pairs

ACTION_DIAL ***tel:123***
Display the phone dialer with the given number filled in.

ACTION_VIEW ***http://www.google.com***
Show Google page in a browser view. Note how the VIEW action does what is considered the most reasonable thing for a particular URI.

ACTION_EDIT ***content://contacts/people/2***
Edit information about the person whose identifier is "2".

ACTION_VIEW ***content://contacts/people/2***
Used to start an activity to display 2-*nd* person.

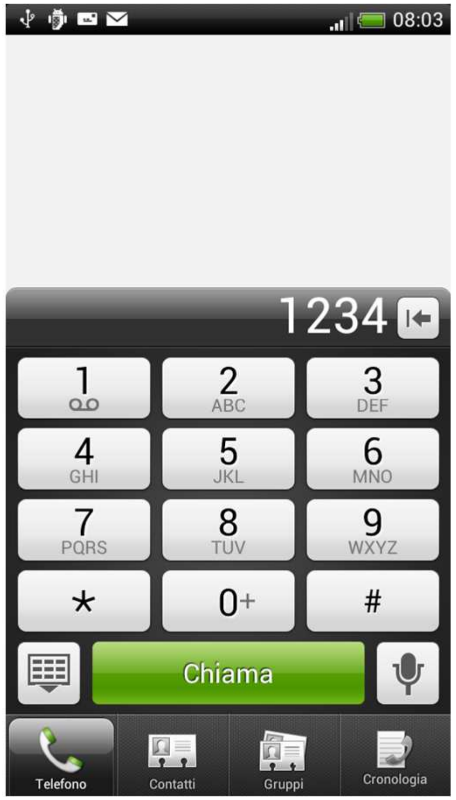
ACTION_VIEW ***content://contacts/* people/**
Display a list of people, which the user can browse through. Selecting a particular person to view would result in a new intent

Example: placing a call

```
Intent intent = new Intent();
intent.setAction(Intent.ACTION_DIAL);
intent.setData(Uri.parse("tel:1234"));
startActivity(intent);
```

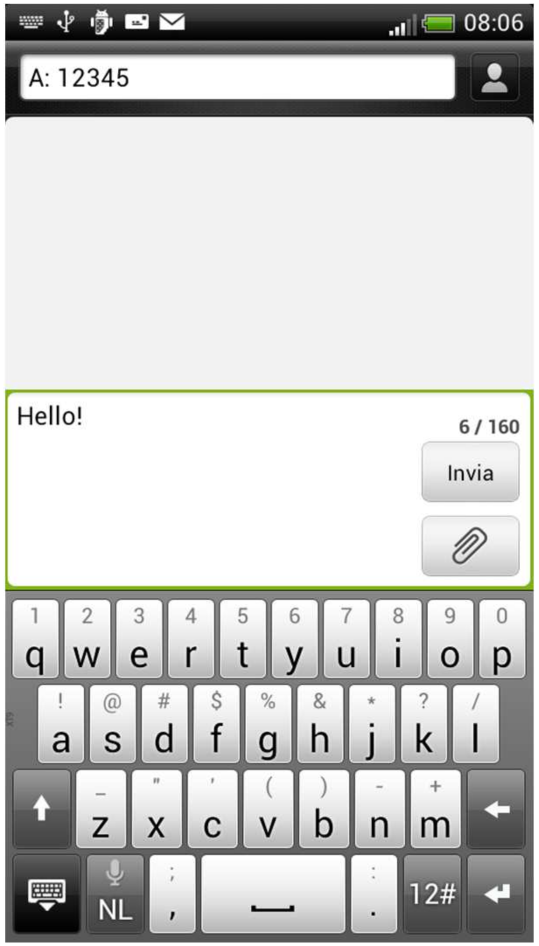
Same as

```
Intent intent = new Intent(Intent.ACTION_DIAL,Uri.parse("tel:1234"));
```



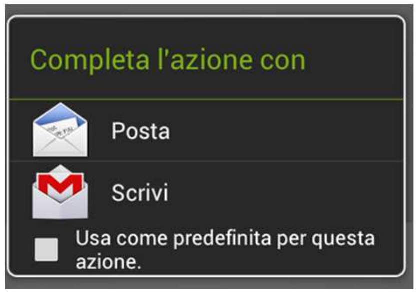
Example: sending sms

```
intent.setAction(Intent.ACTION_SENDTO);
intent.setData(Uri.parse("sms:12345"));
intent.putExtra("sms_body","Hello!");
```



Example: sending an email

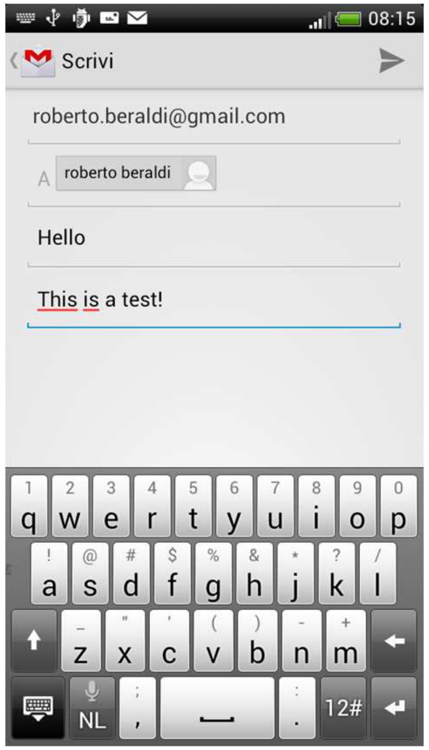
```
intent.setAction(Intent.ACTION_SENDTO);
intent.setData(Uri.parse("mailto:beraldi@dis.uniroma1.it"));
intent.putExtra(Intent.EXTRA_SUBJECT,"Hello");
intent.putExtra(Intent.EXTRA_TEXT,"This is a test!");
```


Completa l'azione con

Posta

Scrivi

Usa come predefinita per questa azione.


Scrivi

roberto.beraldi@gmail.com

A roberto beraldi

Hello

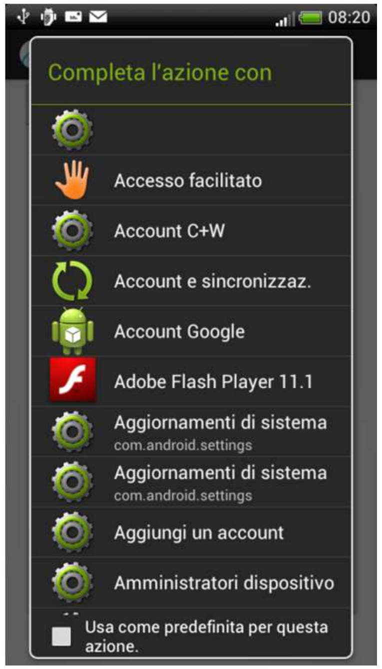
This is a test!

- The are two activities in the device that can perform the action
- The user needs to select one
- Can set the choice as the default

Multiple activities may perform the action

- If there are many Activities that can perform the required action, then the user needs to select one
- In this example, the system proposes all the installed application that declares to be able to respond to the MAIN action

```
intent.setAction(Intent.ACTION_MAIN);
startActivity(intent);
```


08:20

Completa l'azione con

Accesso facilitato

Account C+W

Account e sincronizzaz.

Account Google

Adobe Flash Player 11.1

Aggiornamenti di sistema
com.android.settings

Aggiornamenti di sistema
com.android.settings

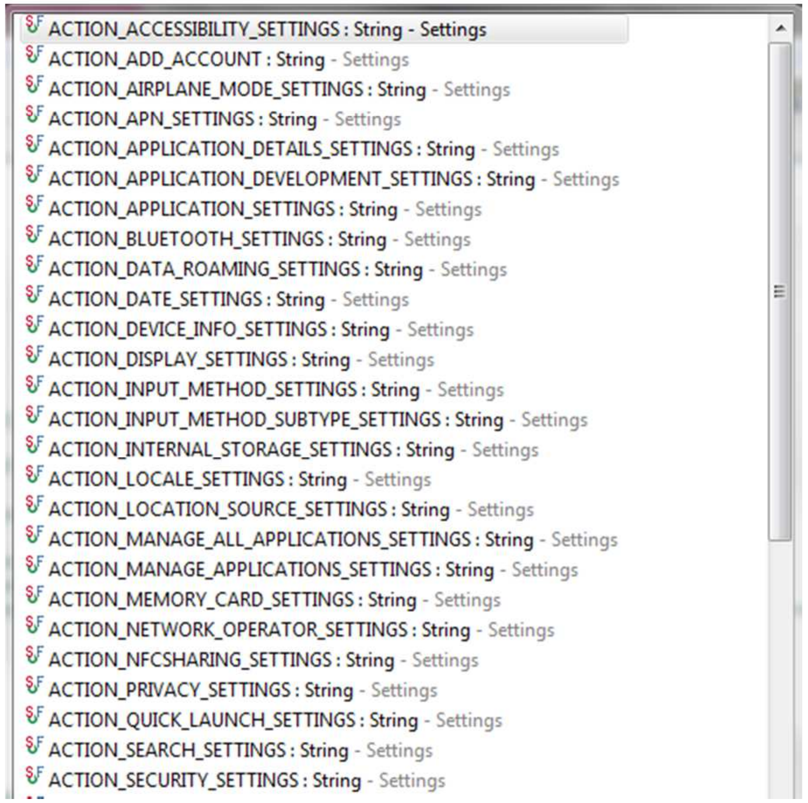
Aggiungi un account

Amministratori dispositivo

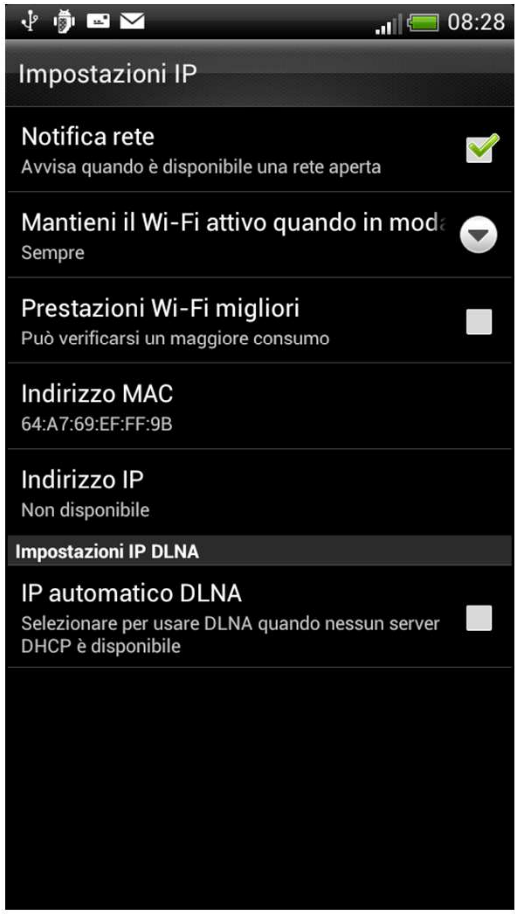
Usa come predefinita per questa azione.

Another example: showing settings

```
intent.setAction(android.provider.Settings.ACTION_WIFI_IP_SETTINGS);
```



```
ACTION_ACCESSIBILITY_SETTINGS : String - Settings
ACTION_ADD_ACCOUNT : String - Settings
ACTION_AIRPLANE_MODE_SETTINGS : String - Settings
ACTION_APN_SETTINGS : String - Settings
ACTION_APPLICATION_DETAILS_SETTINGS : String - Settings
ACTION_APPLICATION_DEVELOPMENT_SETTINGS : String - Settings
ACTION_APPLICATION_SETTINGS : String - Settings
ACTION_BLUETOOTH_SETTINGS : String - Settings
ACTION_DATA_ROAMING_SETTINGS : String - Settings
ACTION_DATE_SETTINGS : String - Settings
ACTION_DEVICE_INFO_SETTINGS : String - Settings
ACTION_DISPLAY_SETTINGS : String - Settings
ACTION_INPUT_METHOD_SETTINGS : String - Settings
ACTION_INPUT_METHOD_SUBTYPE_SETTINGS : String - Settings
ACTION_INTERNAL_STORAGE_SETTINGS : String - Settings
ACTION_LOCALE_SETTINGS : String - Settings
ACTION_LOCATION_SOURCE_SETTINGS : String - Settings
ACTION_MANAGE_ALL_APPLICATIONS_SETTINGS : String - Settings
ACTION_MANAGE_APPLICATIONS_SETTINGS : String - Settings
ACTION_MEMORY_CARD_SETTINGS : String - Settings
ACTION_NETWORK_OPERATOR_SETTINGS : String - Settings
ACTION_NFCSHARING_SETTINGS : String - Settings
ACTION_PRIVACY_SETTINGS : String - Settings
ACTION_QUICK_LAUNCH_SETTINGS : String - Settings
ACTION_SEARCH_SETTINGS : String - Settings
ACTION_SECURITY_SETTINGS : String - Settings
```



08:28

Impostazioni IP

Notifica rete ✓
Avvisa quando è disponibile una rete aperta

Mantieni il Wi-Fi attivo quando in moda
Sempre

Prestazioni Wi-Fi migliori ☐
Può verificarsi un maggiore consumo

Indirizzo MAC
64:A7:69:EF:FF:9B

Indirizzo IP
Non disponibile

Impostazioni IP DLNA

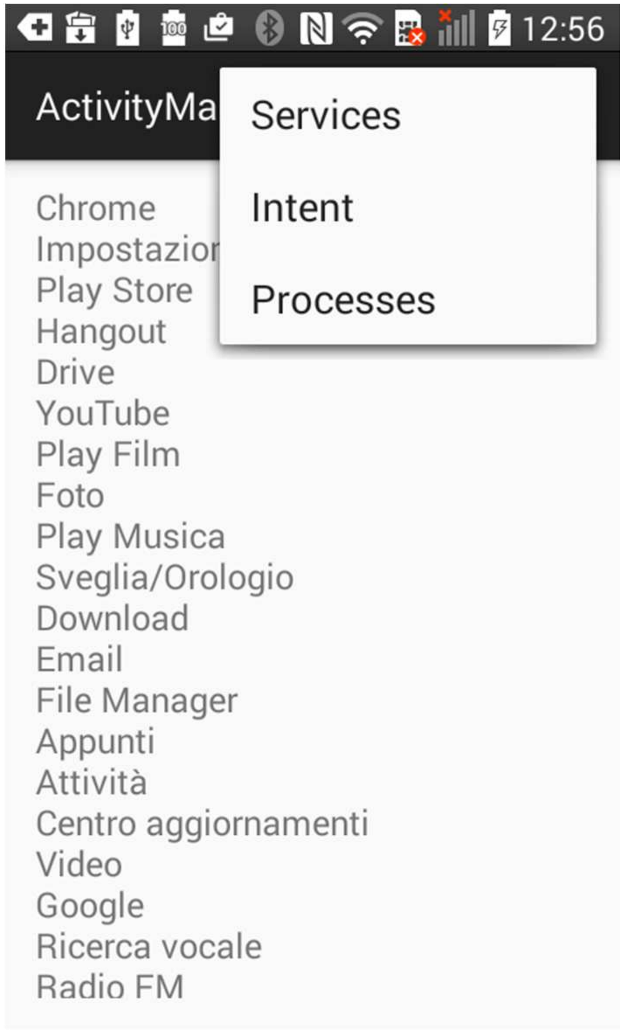
IP automatico DLNA ☐
Selezionare per usare DLNA quando nessun server DHCP è disponibile

Implicit intent resolution

- Action Test
  - The Intent’s action must be declared in the intent-filter
- Category Test
  - All the categories of the Intent must be declared in the intent-filter, not vice versa
- Data Test
  - The data scheme must be declared in intent-filter

- For details see:
- http://developer.android.com/guide/components/intents-filters.html

ActityManager (demo)



Chrome
Impostazior
Play Store
Hangout
Drive
YouTube
Play Film
Foto
Play Musica
Sveglia/Orologio
Download
Email
File Manager
Appunti
Attività
Centro aggiornamenti
Video
Google
Ricerca vocale
Radio FM

Custon action

- User can define its own action
- The action name should go in the manifest file, togheter with category DEFAULT
- Example:

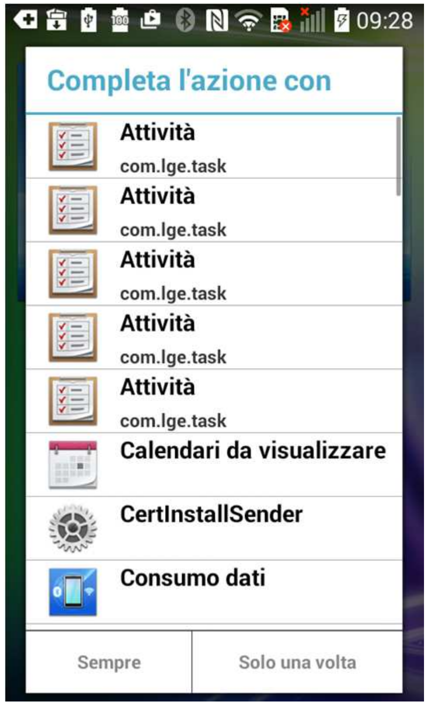
```
<activity android:name=".ThirdActivity">
<intent-filter>
<action android:name="android.intent.action.VIEW2"></action>
<category
android:name="android.intent.category.DEFAULT"></category>
</intent-filter>
```

ActivityCalls (demo)

Sending intent from debugger

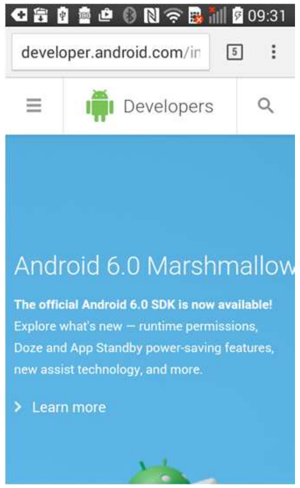
```
adb shell am start -a android.intent.action.VIEW
```

```
C:\Users\roberto\AppData\Local\Android\sdk\platform-tools>
adb shell am start –a android.intent.action.VIEW
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
Starting: Intent { act=android.intent.action.VIEW }
```



Sending intent from the debugger

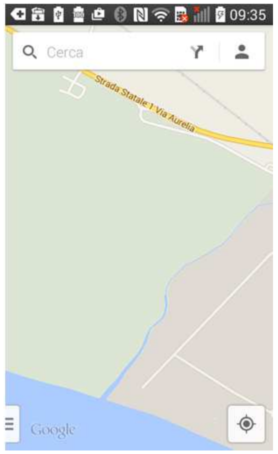
- **adb shell am start –a android.intent.action.VIEW –d "http://developer.android.com"**
- *Starting: Intent { act=android.intent.action.VIEW dat=http://developer.android.com }*



Android 6.0 Marshmallow

The official Android 6.0 SDK is now available! Explore what's new — runtime permissions, Doze and App Standby power-saving features, new assist technology, and more.

- **adb shell am start –a android.intent.action.VIEW -d "geo:42,12"**
- *Starting: Intent { act=android.intent.action.VIEW dat=geo:42,12 }*



Example: Using maps

- It is possible to show google maps or getting driving directions very easily

- intent.setAction(Intent.ACTION_VIEW);intent.setData(Uri.parse("**geo:42,12**"));

- intent.setAction(Intent.ACTION_VIEW);intent.setData(Uri.parse("**http://maps.google.com/maps?sadd=42.12,10.2&daddr=42.12,10.11**"));

-

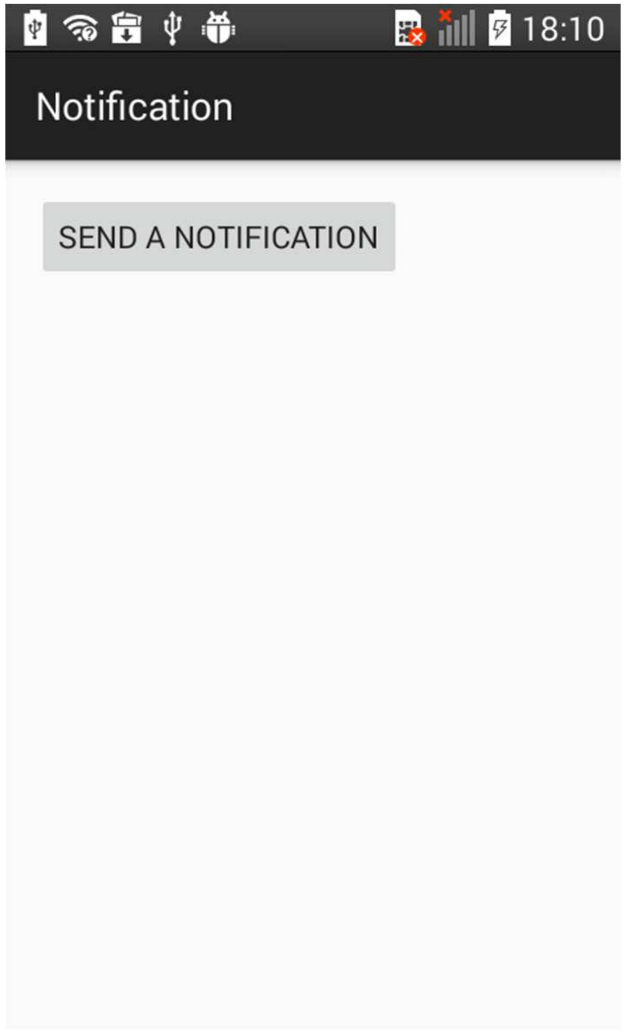
Exercise

- Write a simple activity for typing a phone number and then place the call

PendingIntent

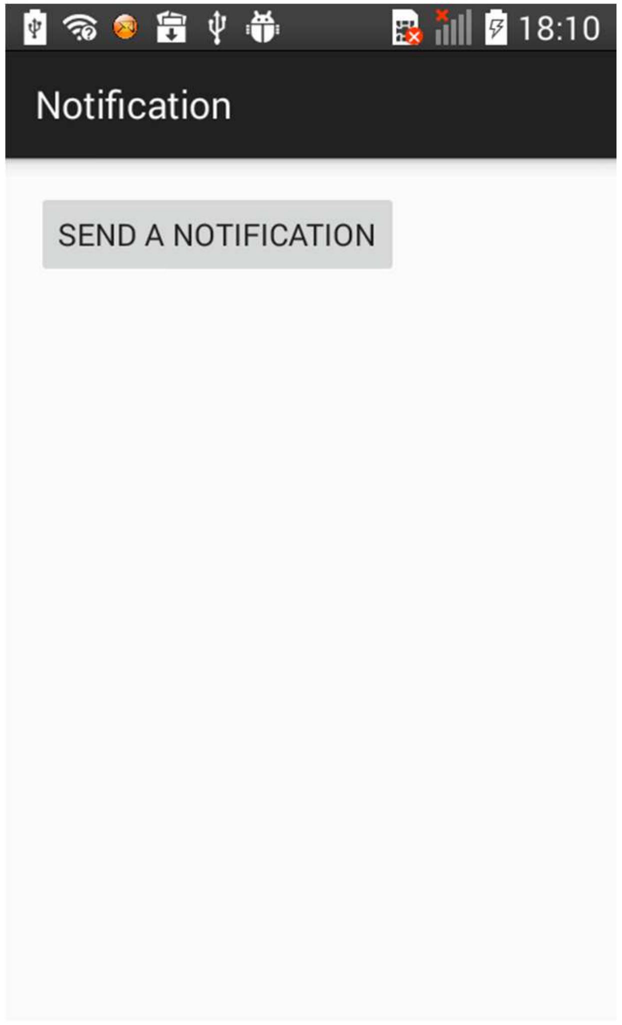
- It specifies an action to take in the future
- The application that will execute that Intent will have the same permissions as the sending application, whether or not such application is still around when the Intent is eventually invoked.
- For example used in notification

Notication (demo)


18:10

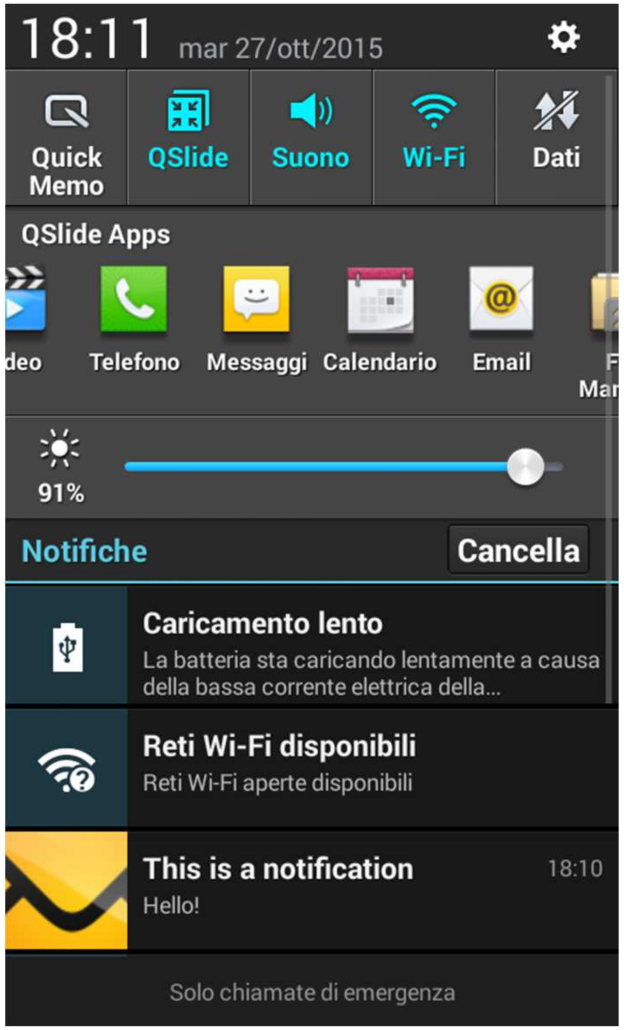
Notification

SEND A NOTIFICATION


18:10

Notification

SEND A NOTIFICATION


18:11 mar 27/ott/2015

Notifiche Cancella

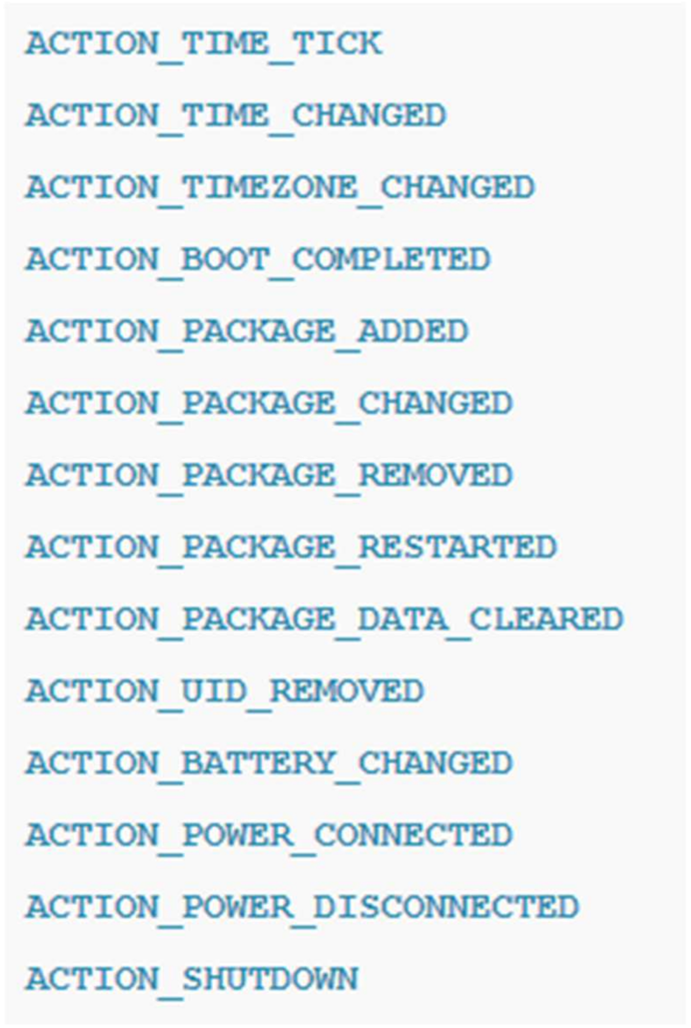
Caricamento lento
La batteria sta caricando lentamente a causa della bassa corrente elettrica della...

Reti Wi-Fi disponibili
Reti Wi-Fi aperte disponibili

This is a notification 18:10
Hello!

Solo chiamate di emergenza

Broadcast intents and receivers



ACTION_TIME_TICK

ACTION_TIME_CHANGED

ACTION_TIMEZONE_CHANGED

ACTION_BOOT_COMPLETED

ACTION_PACKAGE_ADDED

ACTION_PACKAGE_CHANGED

ACTION_PACKAGE_REMOVED

ACTION_PACKAGE_RESTARTED

ACTION_PACKAGE_DATA_CLEARED

ACTION_UID_REMOVED

ACTION_BATTERY_CHANGED

ACTION_POWER_CONNECTED

ACTION_POWER_DISCONNECTED

ACTION_SHUTDOWN

Additional ations are generated by other classes i.e., BluetoothDevice or Managers

Broadcast Receiver

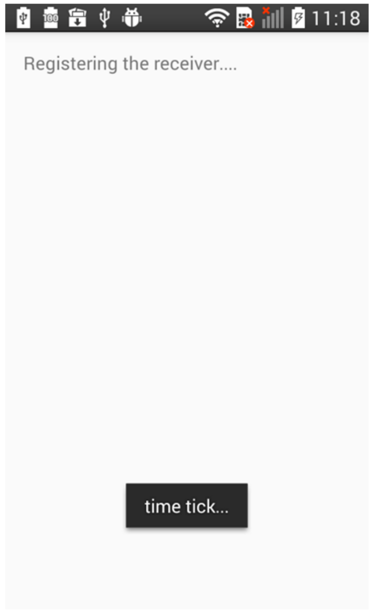
- It’s a software component that reacts to system-wide events.
- Inherits directly from java.lang.Object

Registering Broadcast receivers

- Registration to events can either be
- Statically (through XML, e.g., <receiver> tag)
- Dynamically (from an activity). Called in the UI thread..
- Statically registered receivers reamin dormant and respond to the event
- Dynamically registered event are alive as long as the registering activity is alive
- Subscription to some events can only be done dynamically (e.g., TIME_TICK – this is to avoid battery drain)

BroadcastReceiver (demo)

- Register to the TIME_TICK event
- Warning: the registration can only be done dynamically from the code. Also, for security reason it cannot be generated (sendBroadcast(…))


Registering the receiver....

time tick...

when the time changes a toast is displayed
Useful for example to perform polling…

Example of Broadcast receiver

- BOOT_COMPLETED:
  - Warning: RECEIVE_BOOT_COMPLETED permission is required
- The receiver could for example generate a user notification, start an activity or a service

Sending broadcast intents

- An application can send brodcast intent (some system events are ‘protected’)
- 3 different ways (see documentation for details)
-
- sendBroadcast()
- sendStickyBroacast() (should be avoided)
- sendOrderedBroadcast (priority based reception)