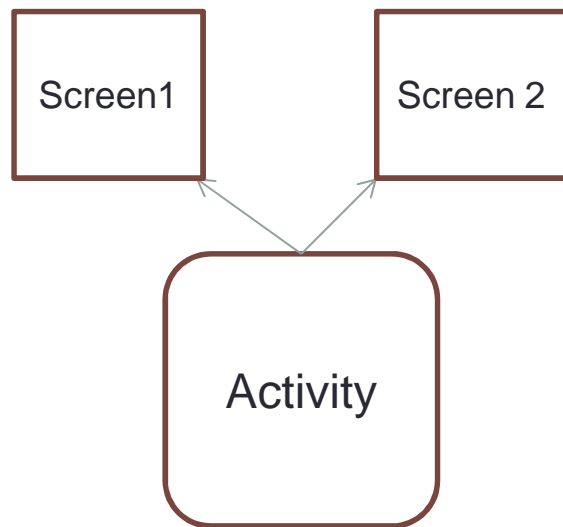




INTENTS

Why using intents



- An activity may manage many layout file (screens)
- Intents, provides a way for an activity to start another activity (thus changing screen)
- Beside this simple usage, there are many other reasons to use intent

Example

```
package com.example.activitywithtwoscreens;

import android.app.Activity;

public class MainActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.screen1);
        Button btn = (Button)findViewById(R.id.button1);

        btn.setOnClickListener(new ClickHandler());
    }

    private class ClickHandler implements OnClickListener {
        public void onClick(View v) {
            setContentView(R.layout.screen2);
        }
    }
}
```

Example

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/button1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Go to Screen 2..." />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="This is Screen1" />

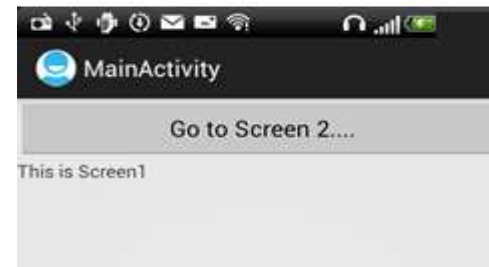
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

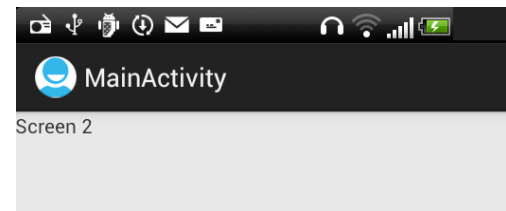
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Screen 2" />

</LinearLayout>
```

HOME



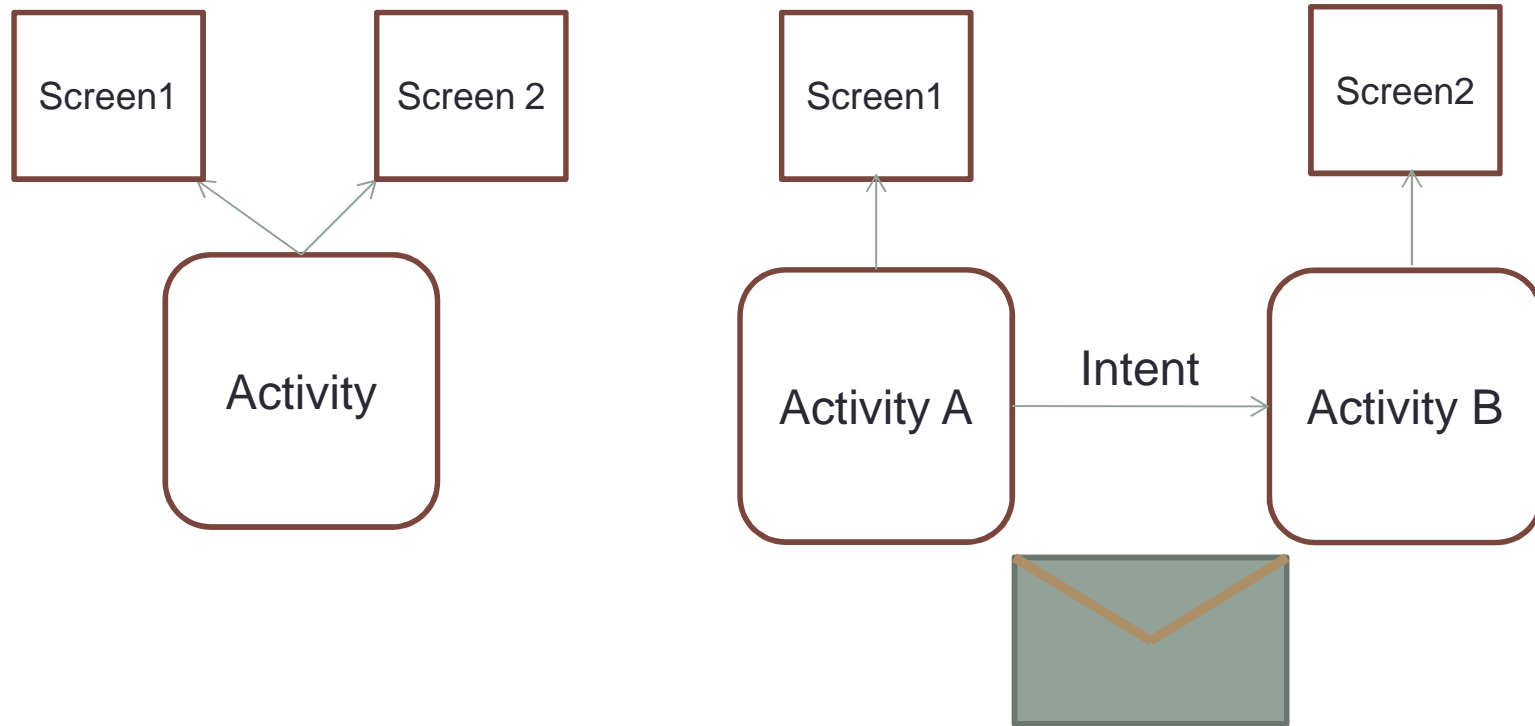
Push



back button

HOME

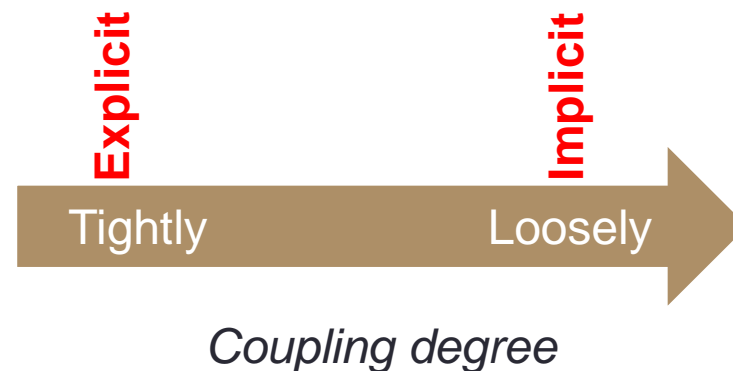
Why using intents



Target Address= Explicit or Implicit
Data to work on
Extra data

Intents

- Intents are the gluing elements between software components, e.g., activities and services
- An Intent may be **explicit** when it exactly starts another activity (or service)
- or **implicit** otherwise: it just specifies the *action* the activity should provide
- It can be **broadcast** when it announces something to all



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

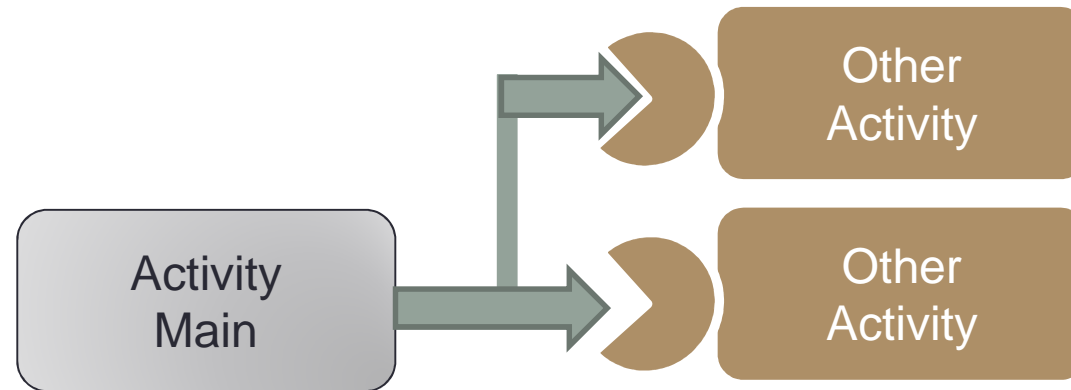
Example

```
public class MainActivity extends Activity  
  
Intent intent = new Intent(MainActivity.this,  
    SecondActivity.class);  
startActivity(intent);
```

```
<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>
```

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

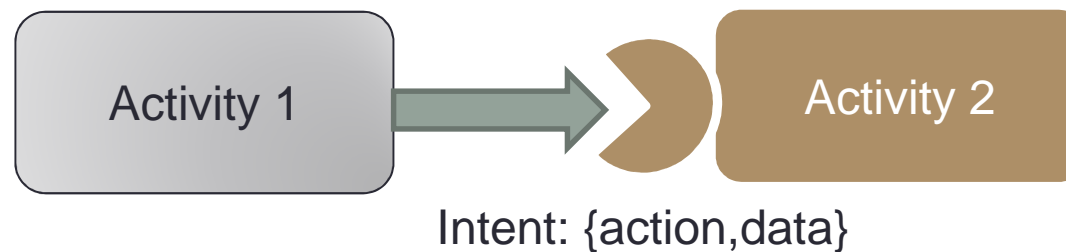

Implicit intents



- 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

Implicit intent

- An implicit Intent has the following arguments:
- **Action**: A built-in action to be performed, like ACTION_VIEW, ACTION_EDIT, etc.
- **Data**: The primary data to operate on (extra data can be added), expressed as a URI, e.g., a phone number



Example of 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

Unicast

- 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

Broadcast

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/*

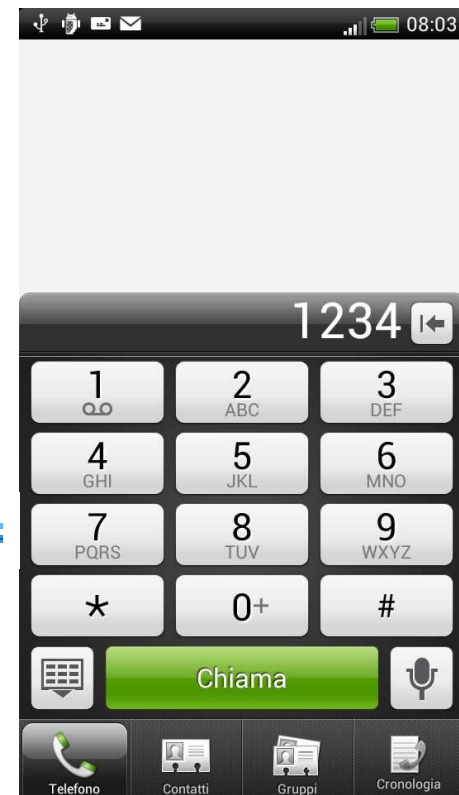
Display a list of people, which the user can browse through. Selecting a particular person to view would result in a new intent

Example: placing a call

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

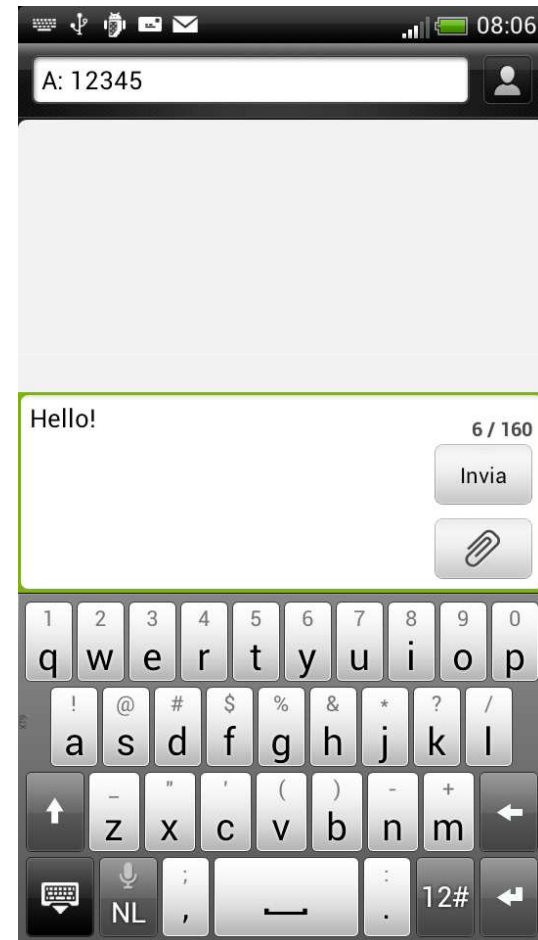
Same as

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



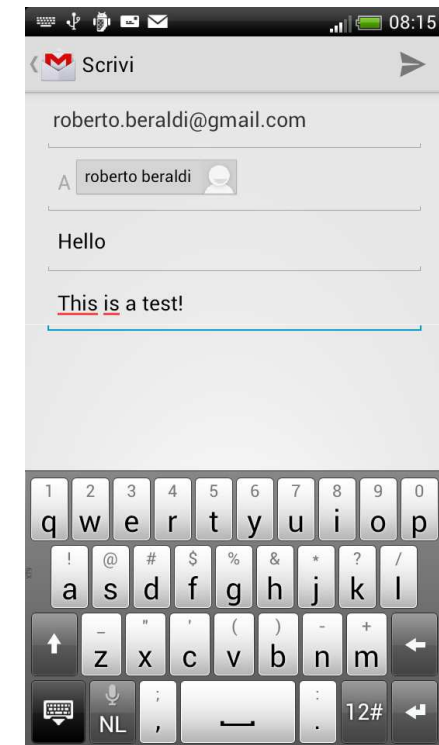
Example: sending sms

```
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!");
```



- There 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

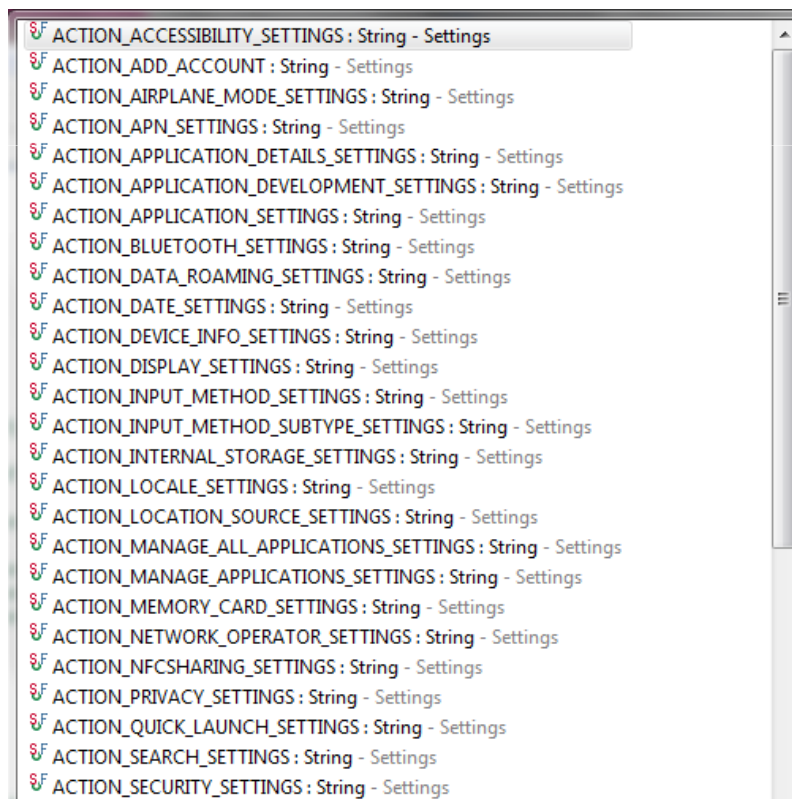
- If there are many Activities that can perform the required action, then the user needs to select one
- In this example, the system proposes all the installed application that declares to be able to respond to the MAIN action

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



Another example: showing settings

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



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 call and then place the call

Category

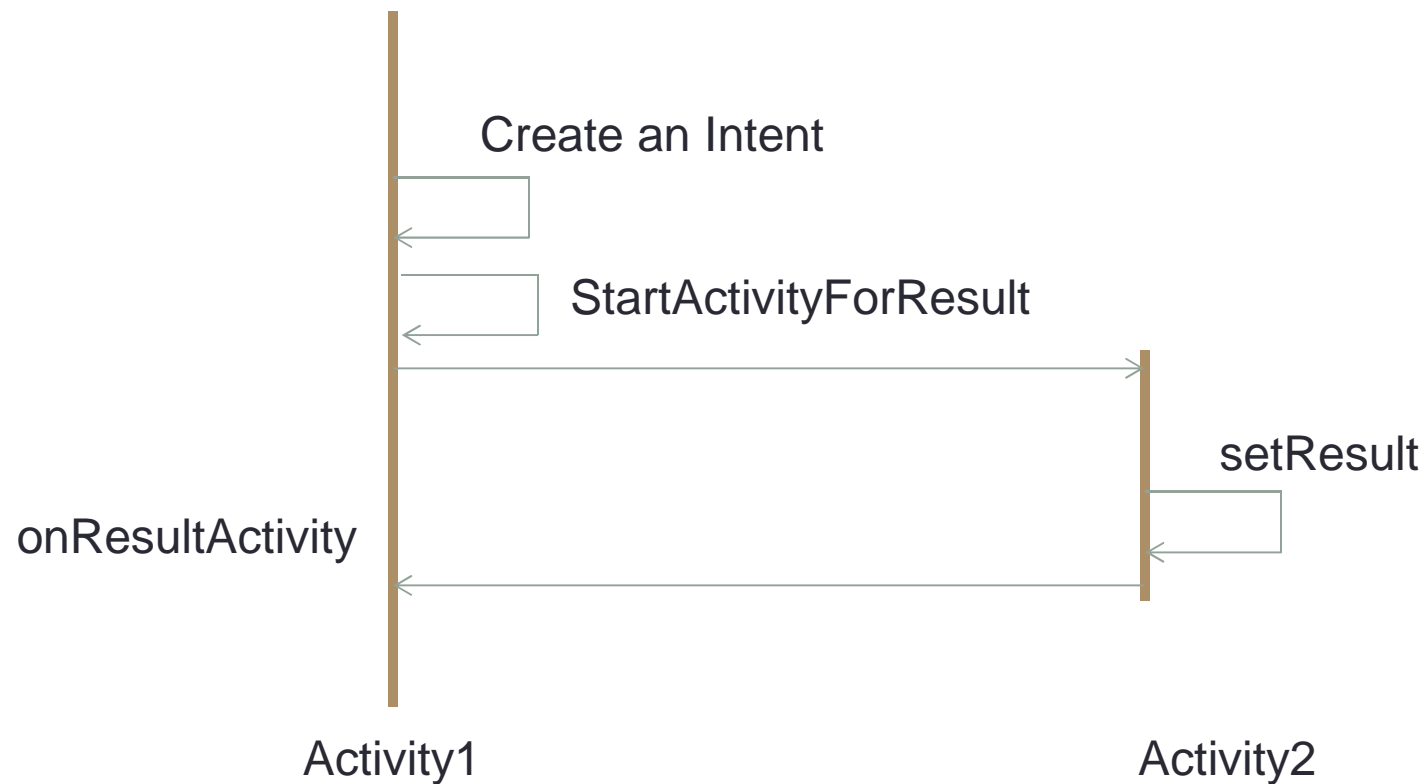
- Specify additional information about what an intent can do

Constant	Meaning
<code>CATEGORY_BROWSABLE</code>	The target activity can be safely invoked by the browser to display data referenced by a link – for example, an image or an e-mail message.
<code>CATEGORY_GADGET</code>	The activity can be embedded inside of another activity that hosts gadgets.
<code>CATEGORY_HOME</code>	The activity displays the home screen, the first screen the user sees when the device is turned on or when the <i>Home</i> button is pressed.
<code>CATEGORY_LAUNCHER</code>	The activity can be the initial activity of a task and is listed in the top-level application launcher.
<code>CATEGORY_PREFERENCE</code>	The target activity is a preference panel.

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



Example: `com.example.intentdemo2b.Activity1`

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)));  
  
    }  
  
}
```

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



Example

- See `ExplicitIntentExample` (from `techotopia`)

Intent filter

- An activity can declare its ability to perform an action, so that it can be activate by other activities in the system
- This is declared in the manifest file in the intent-filter section

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

```
<intent-filter . . . >
  <data android:mimeType="video/mpeg" android:scheme="http" . . . />
  <data android:mimeType="audio/mpeg" android:scheme="http" . . . />
  . . .
</intent-filter>
```