Android: Intent

F. Mallet

Frederic.Mallet@unice.fr

Université Nice Sophia Antipolis

Outline

- Basics
  - Intents: explicit vs. implicit
  - Intent filters
  - startActivity
  - startActivityForResult
- Examples
  - Happy & Sad Smiley
  - MapLocation
  - MapLocationFromContacts

Asynchronous communication

- Intents represent
  - Either operations to be performed (show a map, send message)
    - Start an activity, a service, send a broadcast
  - Or event that has occurred (low battery): see *broadcast receivers*
- An intent is
  - A data structure that carries the message to be sent
    - Action, Data, Category, Type, Component, Extras, Flags
- *Explicit Intent*
  - Know which component should receive the message
- *Implicit Intent*
  - Know which service is required but not the actual provider
    - E.g., Resolve a location, dial a phone number

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

Explicit Intent

- Through the constructor
  - Constructor: Intent(Context c, Class<?> cls)
  - Example:
    - **`Intent i = new Intent(this, SmileyActivity.class)`**
- Through additional methods

  `Intent i = new Intent();`
  - Component

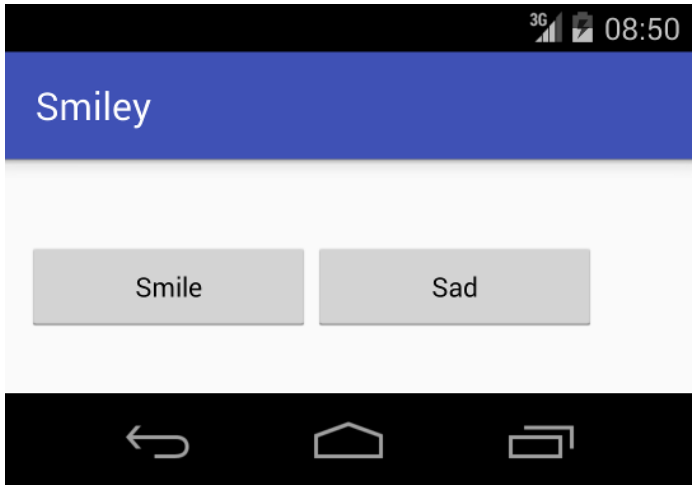
    `i.setComponent(ComponentName c)`
  - Or class

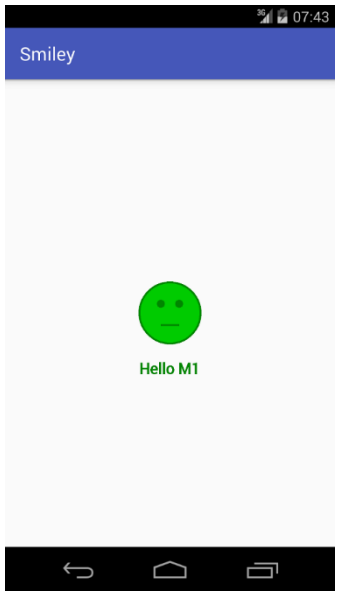
    `i.setClass(this, SmileyActivity.class)`
  - Or class name

    `i.setClassName("fr.unice", "SmileyActivity")`

Example

- Button to start an activity
  - `Button mSmileButton = (Button)findViewById(R.id.smile_button);`
  - ```
    mSmileButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(getApplicationContext(),
                SmileyActivity.class);
            startActivity(i);
        }
    });
    ```



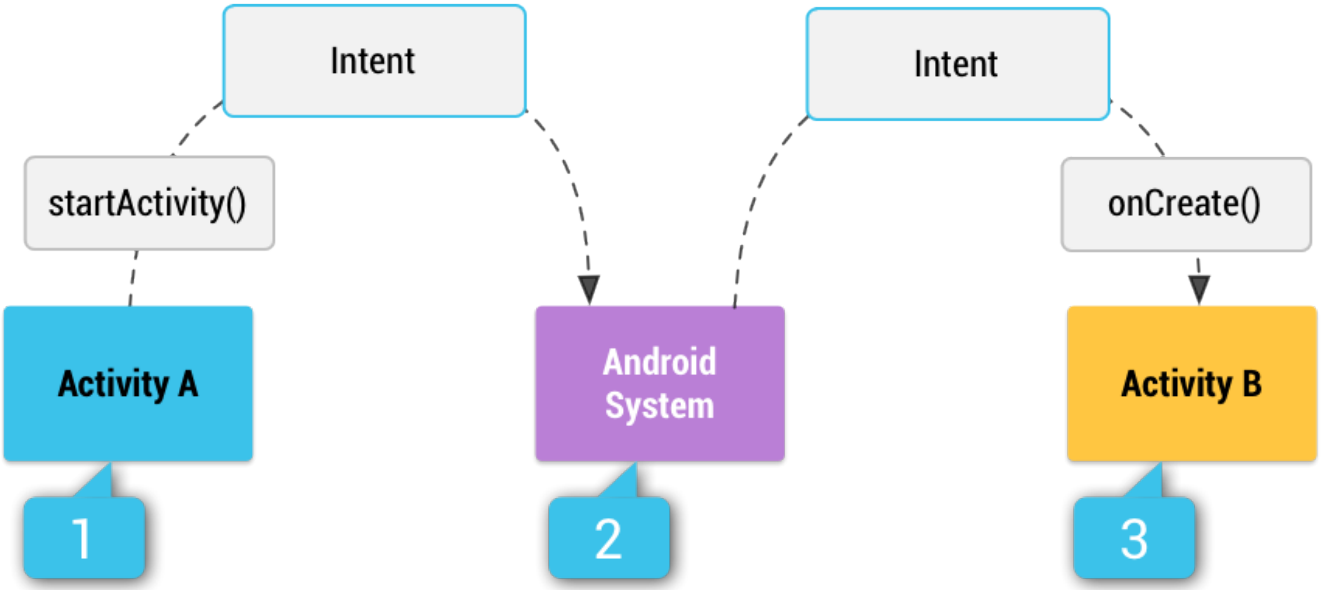


SmileCryActivity

SmileyActivity
    ```

Implicit Intent

- Do NOT define Component or Class
- Define Actions instead
  - A string that represents what is expected
  - Use Intent Filters to pick possible candidates
    - No filter => can only be called by explicit intents



Implicit Intent

- Example

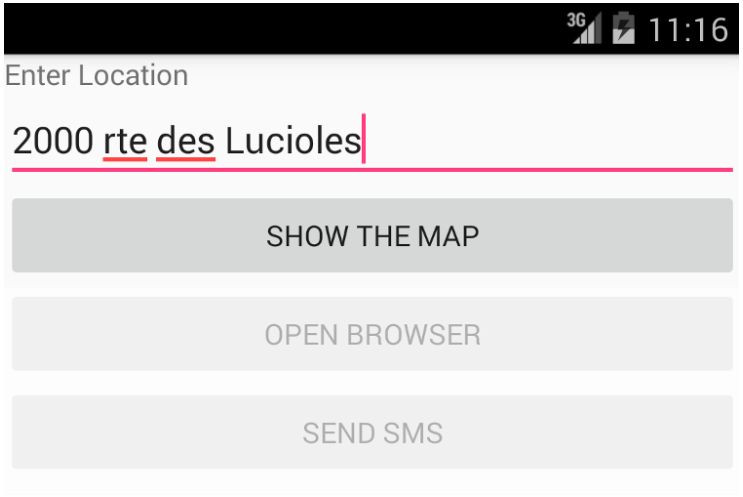
```
String address = addrText.getText().toString();
address = address.replace(' ', '+');
Intent geoIntent = new Intent(Intent.ACTION_VIEW,
                   Uri.parse("geo:0,0?q=" + address));
```

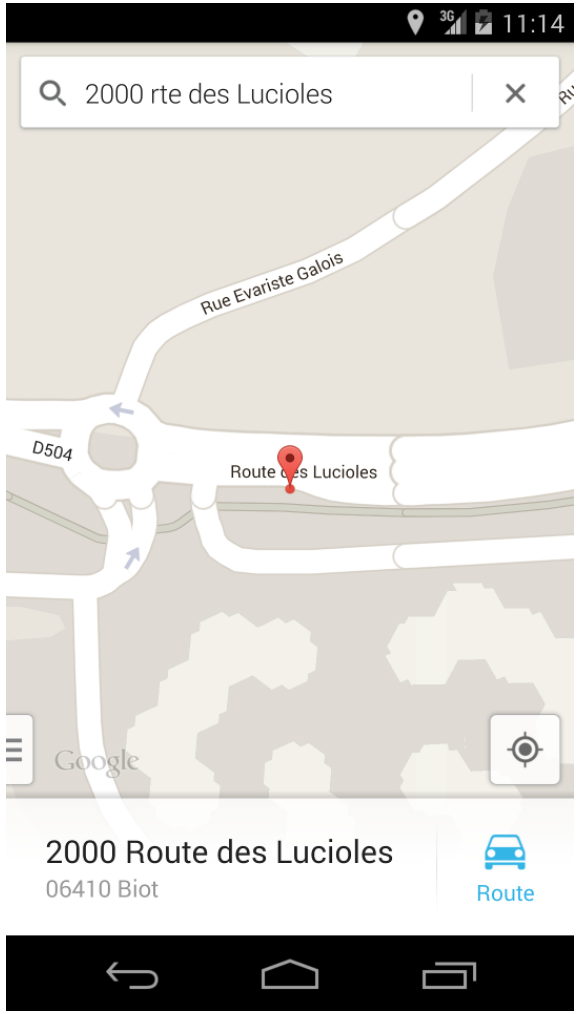
  - Should verify that the implicit intent is resolved

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

Implicit Intent

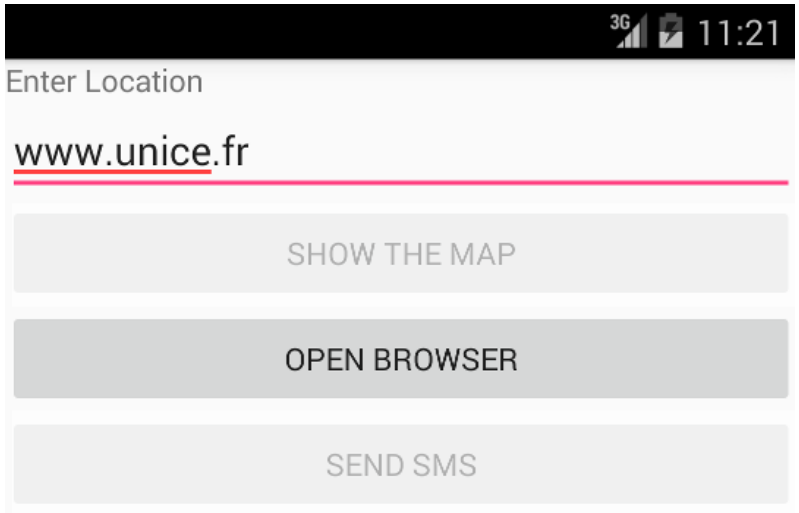
```
new Intent(Intent.ACTION_VIEW, Uri.parse("geo:0,0?q=" + address))
```





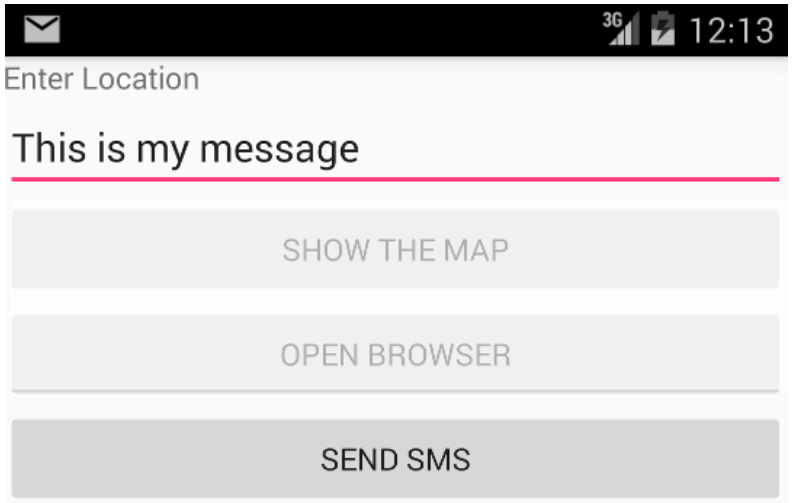
Implicit Intent

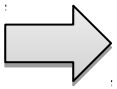
```
new Intent(Intent.ACTION_VIEW, Uri.parse("http://" + address))
```

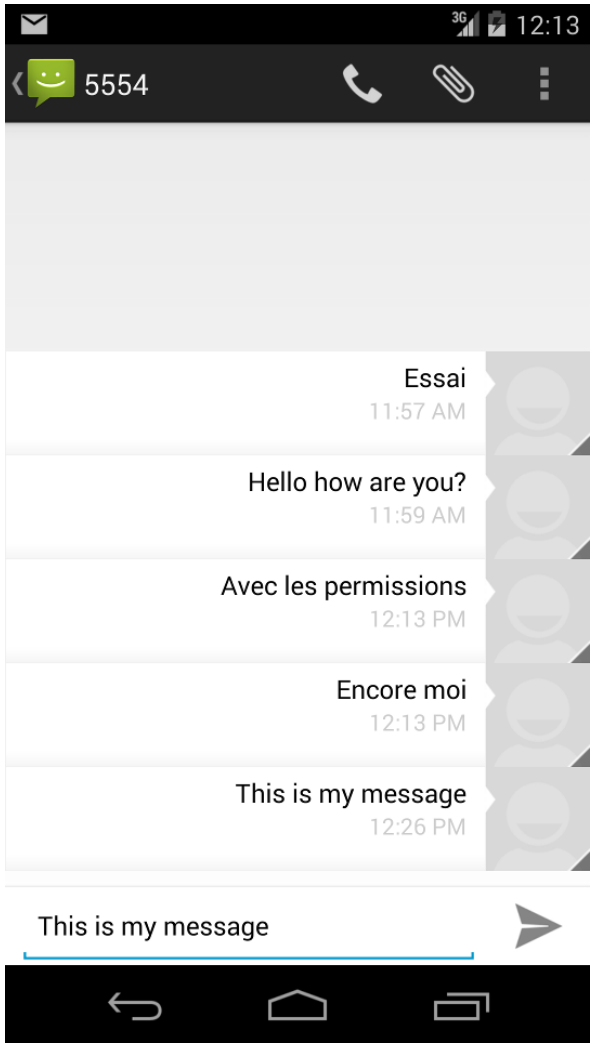


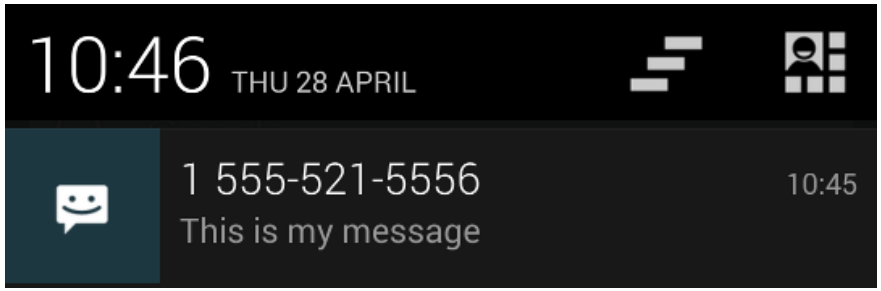
Implicit Intent (with extra data)

```
i = new Intent(Intent.ACTION_VIEW, Uri.parse("sms:5554"))
i.putExtra("sms_body", content);
```

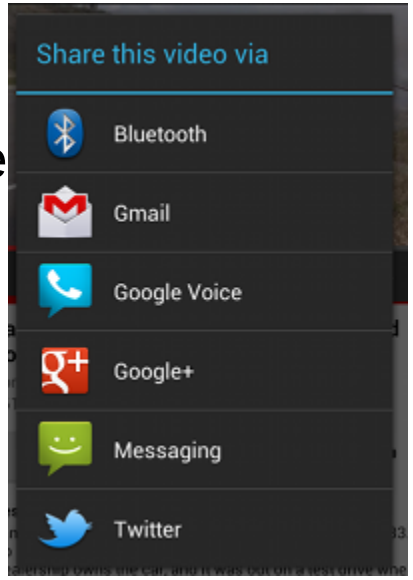








Force a Chooser



- If several components satisfy an implicit intent
  - By default, open a chooser and remember the choice
  - Can also force opening a chooser (no memory)
- Create the intent

```
Intent sendIntent = new Intent(Intent.ACTION_SEND);
```

- Create a chooser for this intent

```
Intent chooser = Intent.createChooser(sendIntent, "choose");
```

- If a selection is done

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

  - Apply the choice made by the chooser

```
    startActivity(chooser);
}
```

Receive an implicit intent

- Should declare an Intent Filter in the Manifest.xml
  - Can filter the action, the data (scheme, host, port, path, MIME)…
    - <scheme>://<host>:<port>/<path>
- Format

```
<intent-filter …>
  <data   android:mimeType="string"
        android:scheme="string"
        android:host="string"
        android:port="string"
        android:path="string"
        android:pathPattern="string"
        android:pathPrefix="string“ />
</intent-filter>
```

Receive an implicit intent

- Should declare an Intent Filter in the Manifest.xml
  - Can filter the action, the data (scheme, host, port, path, MIME)…
    - <scheme>://<host>:<port>/<path>
- Example

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

Extra data

- Key value dictionary
  - Depends on the action, data, type and the receiver
  - Example: ACTION_SEND (Email)

```
intent.putExtra(Intent.EXTRA_MAIL,
    new String[]{xxx@gmail.com, yyy@gouv.fr});
```

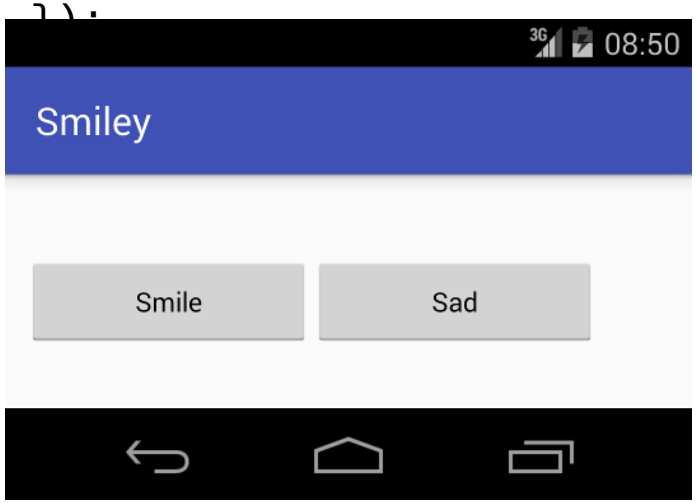
  - ACTION_SENDTO

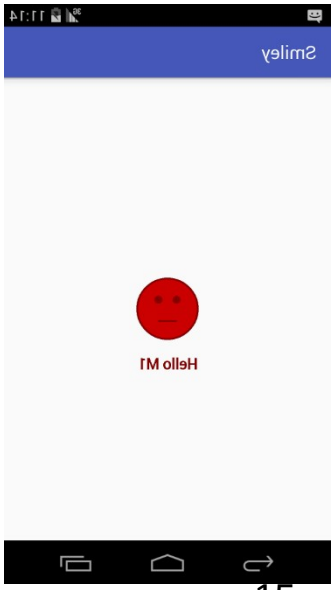
```
intent.putExtra("sms_body", "content of sms");
```

Example

- Button to start an activity
  - `Button mSadButton = (Button)findViewById(R.id.sad_button);`
  - `mSadButton.setOnClickListener(new View.OnClickListener() {`

```
    @Override
    public void onClick(View v) {
        Intent i = new Intent(…, SmileyActivity.class);
i.putExtra(SmileyActivity.BACK_COLOR, Color.argb(255, 200, 0, 0));
i.putExtra(SmileyActivity.FORE_COLOR, Color.argb(255, 128, 0, 0));
        startActivity(i);
    }
});
```





SmileCryActivity

SmileyActivity

Receiving Intents

- Receive intent in onCreate method
  - User getIntent()
  - May extract data, type, extras, flags as necessary
- Example: SmileyActivity

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    Intent i = getIntent();
    int backColor = i.getIntExtra(BACK_COLOR, greenBack);
    int foreColor = i.getIntExtra(FORE_COLOR, greenFore);
    setContentView(new SmileyView(this, backColor, foreColor));
}
```

Receive Intent

Key

Default value

When expecting a result (1/4)

- Some intents require a result
  - The result is received asynchronously (via callback method)
    - Send the intent: **`startActivityForResult`**
    - Receive the intent: **`getIntent`**
    - Return the result: **`setResult`**
    - Receive the result: **`onActivityResult`**
- Send the intent

```
Intent intent = new Intent(Intent.ACTION_PICK,
      ContactsContract.Contacts.CONTENT_URI);
```

```
startActivityForResult(intent, PICK_CONTACT_REQUEST);
```

int: identify the request

When expecting a result (2/4)

- Some intents require a result
  - The result is received asynchronously (via callback method)
    - Send the intent: **`startActivityForResult`**
    - Receive the intent: **`getIntent`**
    - Return the result: **`setResult`**
    - Receive the result: **`onActivityResult`**
- Receive the result

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

    if (resultCode == Activity.RESULT_OK
     && requestCode == PICK_CONTACT_REQUEST) {
         …
    }
```

When expecting a result (3/4)

- Some intents require a result
  - The result is received asynchronously (via callback method)
    - Send the intent: **`startActivityForResult`**
    - Receive the intent: **`getIntent`**
    - Return the result: **`setResult`**
    - Receive the result: **`onActivityResult`**
- Receive the intent

```
Intent intent = getIntent();
Uri data = intent.getData();

if (intent.getType().equals("text/plain")) {
    // Handle intents with text ...
}
int backColor = intent.getIntExtra(BACK_COLOR, greenBack);
```

When expecting a result (4/4)

- Some intents require a result
  - The result is received asynchronously (via callback method)
    - Send the intent: **`startActivityForResult`**
    - Receive the intent: **`getIntent`**
    - Return the result: **`setResult`**
    - Receive the result: **`onActivityResult`**
- Return the result

```
Intent result = new Intent("com.example.RESULT_ACTION",
                    Uri.parse("content://result_uri"));
setResult(Activity.RESULT_OK, result);
finish();
```

COMPLEMENTS

Sending SMS

- Previous example
  - Open the SMS dialog application
    => user needs to confirm SMS
  - This is to avoid requiring permissions
- Actually sending an SMS

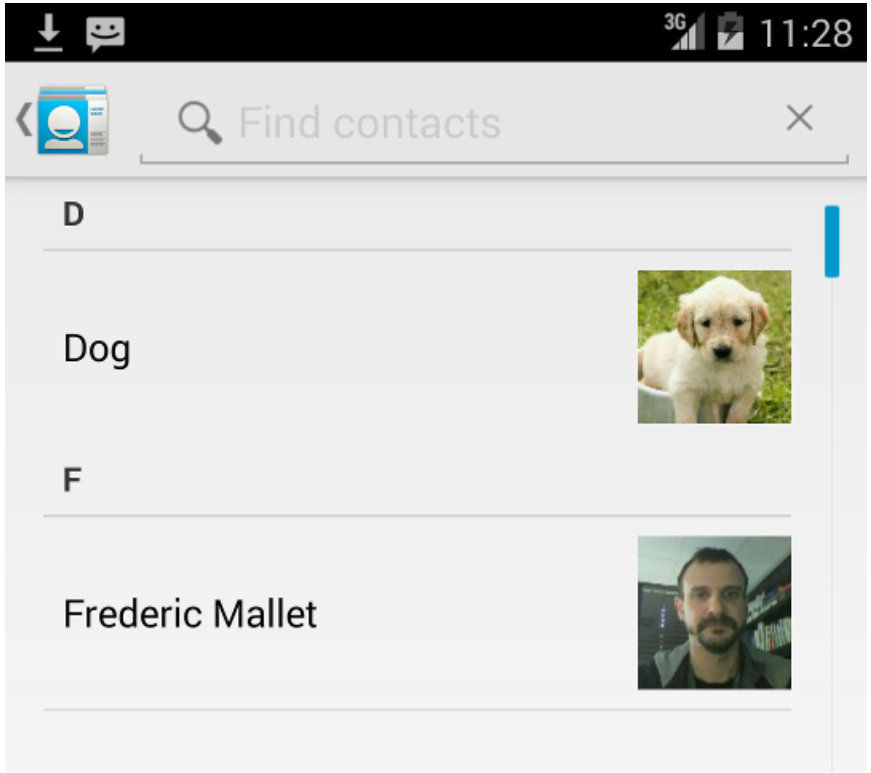
```
SmsManager smsManager = SmsManager.getDefault();
smsManager.sendTextMessage("5554", null, content, null, null);
```

- It requires the right permissions in manifest.xml

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

Pick a contact

```
i = new Intent(Intent.ACTION_PICK,
                                ContactsContract.Contacts.CONTENT_URI);
startActivityForResult(intent, PICK_CONTACT_REQUEST);
```



Extracting contact information

- (see Slide 18)

```
ContentResolver cr = getContentResolver();
Uri dataUri = data.getData();
String[] projection = { ContactsContract.Contacts._ID };
Cursor cursor = cr.query(dataUri, projection, null, null, null);

if (null != cursor && cursor.moveToFirst()) {
   String id = cursor
         .getString(cursor.getColumnIndex(ContactsContract.Contacts._ID));
   String where = ContactsContract.Data.CONTACT_ID + " = ? AND " +
                  ContactsContract.Data.MIMETYPE + " = ?";
   String[] whereParameters = new String[] { id,
     ContactsContract.CommonDataKinds.StructuredPostal.CONTENT_ITEM_TYPE };
   Cursor addrCur = cr.query(ContactsContract.Data.CONTENT_URI, null,
                    where, whereParameters, null);

   if (null != addrCur && addrCur.moveToFirst()) {
      String formattedAddress = addrCur.getString(
                addrCur.getColumnIndex(ContactsContract.CommonDataKinds.
                                       StructuredPostal.FORMATTED_ADDRESS));
```


http://developer.android.com/reference/android/provider/ContactsContract.html

References

- Android Developers
  - http://developer.android.com/training/index.html
- Other lectures at UNS
  - P. Renevier
  - E. Amosse
- Books
  - Beginning Android, M. L. Murphy, APress
  - Android Programming: The Big Nerd Ranch Guide, B. Phillips, B. Hardy, http://www.bignerdranch.com