Android Telephony

Android Smartphone Programming

University of Freiburg

UNI FREIBURG

Matthias Keil / Tim Aicher
Institute for Computer Science
Faculty of Engineering
University of Freiburg

14. Januar 2018

Outline

- Package *android.telephony* provides classes to monitor basic phone information, tools to manipulate phone numbers and to work with SMS[1].
- Most important class is *TelephonyManager*[4].
- Provides access to information about telephony services of the device.
  - For example network type or call state, which indicates idle or ringing.

```
1 Context.getSystemService(Context.
    TELEPHONY_SERVICE).
```

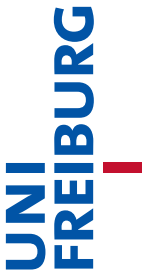
- Very limited access to work with calls for security reasons.
- Possibility to place a call from activity if permissions are set.
- Extend Android manifest:

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

- There are ways to interfere a bit more: Listen to phone state change to ringing and then mute phone.

- How? Create *Intent* with predefined action.

```
1  try {
2     Intent callIntent = new Intent(Intent.
          ACTION_CALL);
3     callIntent.setData(Uri.parse("tel:1234567")
          );
4     startActivity(callIntent);
5  } catch (ActivityNotFoundException e) { ... }
```



- Extend Android manifest:

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

- Use *AudioManager*[2] to manipulate ringer mode.
- Get instance of AudioManager:

```
1 AudioManager Context.getSystemService(Context
      .AUDIO_SERVICE)
```

- Method to change the ringer mode used with parameters *RINGER_MODE_NORMAL*, *RINGER_MODE_SILENT* or *RINGER_MODE_VIBRATE*:

```
1  void AudioManager.setRingerMode(int
       ringerMode)
```

- Class *SmsManager* can be used to send and receive SMS from an application[3].
- Extend Android manifest with needed permissions:
  - android.permission.SEND_SMS
  - android.permission.RECEIVE_SMS
  - android.permission.READ_SMS
  - android.permission.WRITE_SMS

```
1 <uses-permission
2 xmlns:android="http://schemas.android.com/apk
    /res/android"
3 android:name="enter␣permission␣name␣here">
4 </uses-permission>
```

- Receive reference to SmsManager

```
1 static SmsManager SmsManager.getDefault();
```

- Send message

```
1 void SmsManager.sendTextMessage(
2 String destinationAddress,
3 String srcAddress,
4 String text,
5 PendingIntent sentIntent,
6 PendingIntent deliveryIntent);
```

- How to check if a phone number is present in the Contact List?
- Use ContactsContract.PhoneLookup

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

```
1   Uri uri = Uri.withAppendedPath(PhoneLookup.
        CONTENT_FILTER_URI, Uri.encode(
        phoneNumber));
2   Cursor cur = context.getContentResolver().
        query(uri, new String[]{PhoneLookup.
        DISPLAY_NAME, ...);
```

Summary

- Functionality to monitor phone information included in class *TelephonyManager*.
- Application can place calls, but not much more for security reasons.
- Muting phone when phone state changes to ringing allows more interference.
- *SmsManager* allows sending of SMS.

Bibliography

Android Developers.
android.telephony.
http://developer.android.com/reference/android/telephony/package-summary.html.

Android Developers.
AudioManager.
http://developer.android.com/reference/android/media/AudioManager.html.

Android Developers.
SmsManager.
http://developer.android.com/reference/android/telephony/SmsManager.html.

Android Developers.
TelephonyManager.
http://developer.android.com/reference/android/telephony/TelephonyManager.html.

Android Developers.
WallpaperService.
http://developer.android.com/reference/android/service/wallpaper/WallpaperService.html.