

# Android Telephony

## Introduction to Android Smartphone Programming

University of Freiburg

**Matthias Keil**  
Institute for Computer Science  
Faculty of Engineering  
University of Freiburg

31. Januar 2013



**UNI  
FREIBURG**



1 Telephony

2 SMS

3 Calendar

4 Summary



- Package *android.telephony* provides classes to monitor basic phone information, tools to manipulate phone numbers and to work with SMS<sup>[1]</sup>.
- Most important class is *TelephonyManager*<sup>[4]</sup>.
- 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*<sup>[2]</sup> 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<sup>[3]</sup>.
- 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);
```



- To access the built-in calendar the Android manifest has to be extended<sup>[7]</sup>:

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

- Data can be accessed through a Content Provider and a Cursor.
- The Content Provider for the calendar has the following URI:

```
1 content://calendar/calendars
```





- 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.




 **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>.

 **ANDROID DEVELOPERS.**  
**WallpaperService.Engine.**  
<http://developer.android.com/reference/android/service/wallpaper/WallpaperService.Engine.html>.

 [\[DEVELOPER.COM.](#)

