Android: System Services

http://developer.android.com/guide/components/services.html

http://developer.android.com/guide/topics/sensors/sensors_overview.html

Ferruccio Damiani

Università di Torino
www.di.unito.it/~damiani

Mobile Device Programming
(Laurea Magistrale in Informatica, a.a. 2018-2019)

Outline

Outline

Power Services

- The battery of our phones must be used wisely

Example

```
1 val pm = getSystemService(Context.POWER_SERVICE) as PowerManager
2 val wl = pm.newWakeLock(PowerManager.PARTIAL_WAKE_LOCK, "MyApp:MyTag")
3 wl.acquire()
4 // ...screen will stay on during this section...
5 wl.release()
```

- Proper permissions in application's manifest
- Methods: isScreenOn, isPowerSaveMode, ...
- Device battery life will be significantly affected by the use of this API
  - Do not acquire locks unless you really need them
  - Be sure to release them as soon as possible.

Flag Value	CPU	Screen	Keyboard
PARTIAL_WAKE_LOCK	On*	Off	Off

Outline

Vibrator Services

Example

```
1 val vibrator = getSystemService(Context.VIBRATOR_SERVICE) as Vibrator
```

- Some methods:
  - hasVibrator()
  - cancel();
  - vibrator.vibrate(VibrationEffect.createOneShot(milliseconds,amplitude))
- Needs android.permission.VIBRATE

Outline

Alarm Services

- Allows one to schedule an application to be run in the future
  - When an alarm goes off, the Intent that had been registered for it is broadcast by the system and the target application is run automatically
  - Registered alarms are retained while the device is asleep (and can optionally wake it up)
    - They are cleared if it is turned off and rebooted
- Beginning with API 19, the OS shifts alarms to minimize wakeups and battery use
  - There are new APIs to support applications that need strict delivery guarantees

Alarm Services

Example

```
1 val alarmManager = getSystemService(Context.ALARM_SERVICE) as AlarmManager
```

- set(Exact)(type: Int, triggerAtMillis: Long, operation: PendingIntent!)
  - Schedule an allarm
- set(Inexact)Repeating(type: Int, triggerAtMillis: Long, intervalMillis: Long, operation: PendingIntent!)
  - Schedule a repeating alarm
- cancel(operation: PendingIntent!)
  - Remove all alarms that match
- setWindow(type: Int, windowStartMillis: Long, windowLengthMillis: Long, operation: PendingIntent!)
  - Schedule an alarm to be delivered within a given window of time

Outline

Sensors

- Android supports three broad categories of sensors
  - Motion sensors: accelerometers, gravity sensors, gyroscopes, and rotational vector sensors
  - Environmental sensors: barometers, photometers, and thermometers
  - Position sensors: orientation sensors and magnetometers
- Sensor framework helps
  - Determine which sensors are available on a device
  - Determine an individual sensor's capabilities
  - Acquire raw sensor data
  - Register and unregister sensor event listeners that monitor sensor changes
- Key elements
  - SensorManager, Sensor, SensorEvent, and SensorEventListener

Sensors [http://developer.android.com/guide/topics/sensors/sensors_overview.html]

Sensor	Type	Description	Common Uses
TYPE_ACCELEROMETER	Hardware	Measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y, and z), including the force of gravity.	Motion detection (shake, tilt, etc.).
TYPE_AMBIENT_TEMPERATURE	Hardware	Measures the ambient room temperature in degrees Celsius (°C). See note below.	Monitoring air temperatures.
TYPE_GRAVITY	Software or Hardware	Measures the force of gravity in m/s^2 that is applied to a device on all three physical axes (x, y, z).	Motion detection (shake, tilt, etc.).
TYPE_GYROSCOPE	Hardware	Measures a device's rate of rotation in rad/s around each of the three physical axes (x, y, and z).	Rotation detection (spin, turn, etc.).
TYPE_LIGHT	Hardware	Measures the ambient light level (illumination) in lx.	Controlling screen brightness.
TYPE_LINEAR_ACCELERATION	Software or Hardware	Measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y, and z), excluding the force of gravity.	Monitoring acceleration along a single axis.
TYPE_MAGNETIC_FIELD	Hardware	Measures the ambient geomagnetic field for all three physical axes (x, y, z) in μT.	Creating a compass.
TYPE_ORIENTATION	Software	Measures degrees of rotation that a device makes around all three physical axes (x, y, z). As of API level 3 you can obtain the inclination matrix and rotation matrix for a device by using the gravity sensor and the geomagnetic field sensor in conjunction with the `getRotationMatrix()` method.	Determining device position.

Sensor Services

Example

```
1 val deviceSensors: List<Sensor> = mSensorManager.getSensorList(Sensor.TYPE_ALL)
```

- getDefaultSensor() returns default sensor of a given type
  - If a default sensor does not exist, the method call returns null, which means the device does not have that type of sensor

Example

```
1 if (mSensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY) != null) {
2     val gravSensors: List<Sensor> = mSensorManager.getSensorList(Sensor.TYPE_GRAVITY)
3     // Use the version 3 gravity sensor.
4     mSensor = gravSensors.firstOrNull { it.vendor.contains("Google LLC") && it.version == 3 }
5 }
```

How to monitor sensor events

Implement two callback methods that are exposed through the SensorEventListener interface:

- onAccuracyChanged()
  - Accuracy is represented by one of four status constants
    - SENSOR_STATUS_ACCURACY_LOW, SENSOR_STATUS_ACCURACY_MEDIUM, SENSOR_STATUS_ACCURACY_HIGH or SENSOR_STATUS_UNRELIABLE
- onSensorChanged()
  - A SensorEvent object contains information about the new sensor data
    - Accuracy of the data, the sensor that generated the data, the timestamp at which the data was generated, and the new data that the sensor recorded

Outline

Connectivity Service

- Checks the state of network connectivity

Example

```
1 val connectivityManager = getSystemService(Context.CONNECTIVITY_SERVICE) as ConnectivityManager
```

- Monitors Wi-Fi, GPRS, UMTS, ecc.
- Sends broadcast intents when network connectivity changes
- Attempts to "fail over" to another network when connectivity to a network is lost
- Provides an API that allows applications to query the state of the available networks
- Needs proper permissions

Outline

Wi-Fi Services

Manages all aspects of Wi-Fi connectivity

- getWifiState()
  - Returns WIFI_STATE_DISABLED, WIFI_STATE_DISABLING, WIFI_STATE_ENABLED, WIFI_STATE_ENABLING, WIFI_STATE_UNKNOWN
- isWifiEnabled() / setWifiEnabled()
- getConfiguredNetworks()
- addNetwork(WifiConfiguration config)
- updateNetwork(WifiConfiguration config)
- removeNetwork(int netid)
- startScan()
- getScanResults()

Outline

Many other services...

[http://developer.android.com/reference/android/content/Context.html]

For example:

- Audio Service
- Bluetooth Service
- KeyGuard Services
- NFC Service
- Telephony Services
- and other ...

Summary

Android Applications are structured as a single Activity or a group of Activities

- Intents to call other activities
- Layout and Views to setup the GUI
- Events to manage the interactions with the user

Activities execute only in foreground

- Updates in background mode
  - A Service provides a robust environment for background tasks
- Notifications in case of changes