



Développement d'Applications Mobiles sous Android

Les Capteurs

Encadré par :
KHALIFI Hamid

Présenté par :
KIHEL Mohammed
LAASIRI Meryem



● **Capteur**

- Types de capteurs gérés par Android
- Framework (API) d'Android pour gestion des **capteurs**

● **Utilisation des capteurs**

- Agir par rapport aux événements d'un capteur

● **Exemple : Accéléromètre**

- Utilisation du filtre Google Play pour configurer une application avec **capteur**

● **Démo**



Capteur



Types de capteurs gérés par Android :

01

Capteurs de mouvements

02

Capteurs d'environnement

03

Capteurs de position

Capteur	Type	Description	Utilisation
TYPE_ACCELEROMETER (Capteur de mouvements)	Hardware	Mesure l'accélération force en m/s ² appliquée à un appareil sur les trois physiques axes (x, y et z), y compris la force de gravité.	Détection de mouvement (secouer, incliner, etc.).
TYPE_AMBIENT_TEMPERATURE (Capteur d'environnement)	Hardware	Mesure la pièce ambiante température en degrés Celsius (°C). Voir note ci-dessous	Surveillance de l'air températures.
TYPE_ORIENTATION (Capteurs de position)	Software	Mesure les degrés de rotation d'un appareil fait autour des trois axes physiques (x, y,z).	Détermination de positionnement de l'appareil.

Capteur	Android 4.0	Android 2.3	Android 2.2	Android 1.5
TYPE_ACCELEROMETER	Yes	Yes	Yes	Yes
TYPE_AMBIENT_TEMPERATURE	Yes	n/a	n/a	n/a
TYPE_ORIENTATION	Yes	Yes	Yes	Yes
TYPE_LIGHT	Yes	Yes	Yes	Yes
TYPE_PROXIMITY	Yes	Yes	Yes	Yes
TYPE_GRAVITY	Yes	Yes	n/a	n/a



Les principales Classes de l'API

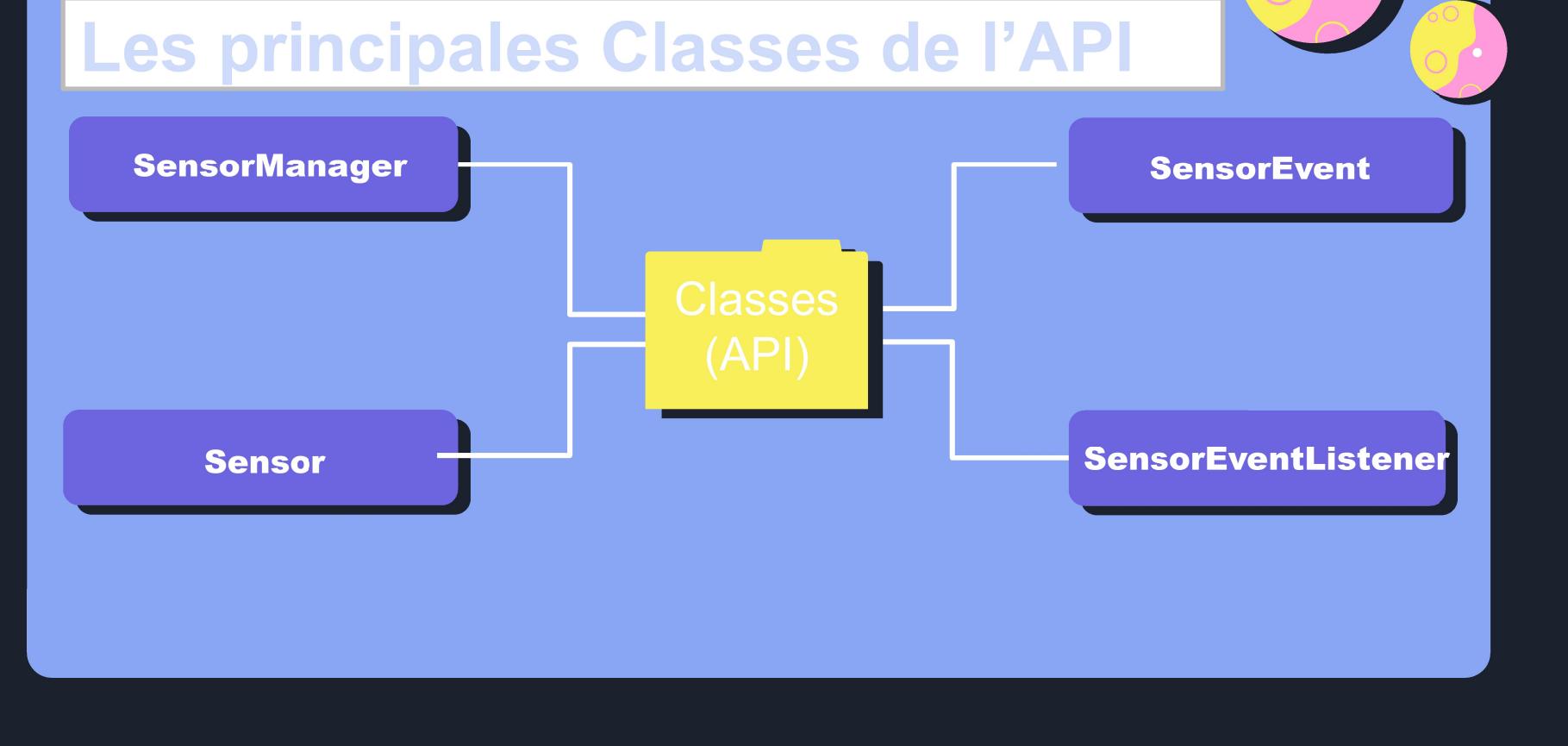
SensorManager

SensorEvent

Sensor

SensorEventListener

**Classes
(API)**





Utilisation des capteurs : Les étapes

01

Récupérer une instance du
service de gestion
des capteurs



Utilisation des capteurs : Les étapes

02

Vérifier si le capteur qui nous intéresse est disponible sur l'appareil



Utilisation des capteurs : Les étapes

03

Si le capteur en question existe, récupérer un objet qui représente le capteur qui nous intéresse



Utilisation des capteurs : Les étapes

04

S'abonner aux événements
du capteur qui nous
intéresse



Utilisation des capteurs : Les étapes

05

Utiliser les informations de l'objet, et encapsuler les données reçues du capteur et faire une action



1

Récupérer une instance du service de gestion des capteurs

2

Lister les capteurs présents sur un appareil

3

Déterminer la présence d'un capteur d'un type donné sur un appareil

//





Récupérer une instance du service de gestion des capteurs

```
private SensorManager mSensorManager;  
...  
mSensorManager = (SensorManager)  
getSystemService(Context.SENSOR_SERVICE);
```



Lister les capteurs présents sur un appareil

```
final SensorManager sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);

List<Sensor> sensorsList =
sensorManager.getSensorList(Sensor.TYPE_ALL);
```



Lister les capteurs présents sur un appareil

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    sensorManager = (SensorManager)  
        getSystemService(SENSOR_SERVICE);  
    listSensor();  
}
```



Lister les capteurs présents sur un appareil

```
private void listSensor() {  
    List<Sensor> sensors =  
        sensorManager.getSensorList(Sensor.TYPE_ALL);  
    StringBuffer sensorDesc = new StringBuffer();
```



Lister les capteurs présents sur un appareil

```
for (Sensor sensor : sensors) {  
  
    sensorDesc.append("New sensor detected : \r\n");  
    sensorDesc.append("\tName: " + sensor.getName() + "\r\n");  
    sensorDesc.append("\tType: " + getType(sensor.getType()) +"\r\n");  
    .....  
}  
.....
```



Lister les capteurs présents sur un appareil

```
Toast.makeText  
(  
    this, sensorDesc.toString(), Toast.LENGTH_LONG  
).show();  
}
```



Déterminer la présence d'un capteur d'un type donné sur un appareil

```
Sensor defaultProximitySensor =  
sensorManager.getDefaultSensor(  
Sensor.TYPE_PROXIMITY  
);  
Log.v(  
"SensorActivity",  
"defaultProximitySensor = " +  
defaultProximitySensor.getName()  
);
```

getDefaultSensor()

Si l'appareil possède plus d'un capteur du même type, il doit être désigné comme capteur par défaut.



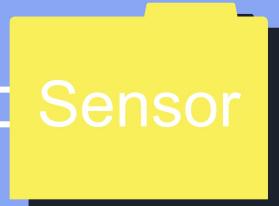
Déterminer les propriétés d'un capteur

`getPower()`

`getMaximumRange()`

`getResolution()`

`getVersion()`





1

S'enregistrer auprès du service comme
écouteur d'événements

2

Implémenter deux méthodes de
callback définies dans l'interface

//





```
@Override  
protected void onResume() {  
    super.onResume();  
    mSensorManager.registerListener(  
        this,  
        mSensor,  
        SensorManager.SENSOR_DELAY_NOR  
        MAL  
    );}
```

registerListener()

S'enregistrer auprès du service
comme écouteur d'événements



Agir par rapport aux événements d'un capteur

```
@Override  
protected void onPause() {  
    super.onPause();  
    mSensorManager.unregisterListener(  
        this  
    );  
}
```

unregisterListener()

S'enregistrer auprès du service
comme écouteur d'événements



Agir par rapport aux événements d'un capteur

```
@Override  
public void onAccuracyChanged  
(Sensor sensor,int accuracy)  
{  
accuracy =  
SensorManagerSENSOR_STATUS_AC  
CURACY_HIGH ;  
Log.d("Sensor",sensor.getType()  
+": "+accuracy); }
```

**onAccuracyChanged(
)**

Implémenter cette méthodes de
callback définie dans l'interface



Agir par rapport aux événements d'un capteur

```
@Override  
public void  
onSensorChanged(SensorEvent  
sensorEvent) {  
    float a = sensorEvent.values[0];  
    .....  
    .....  
}
```

onSensorChanged()

Implémenter cette méthodes de
callback définie dans l'interface

Nom	Dimension du vecteur	Unité	Sémantique	Valeurs[]
Accelerometer	3	m/s ²	Mesure de l'accélération (gravité incluse)	[0] axe x [1] axe y [2] axe z
Light	1	Lux	Mesure de la luminosité	[0]valeur
Proximity	1	mètre	Mesure la distance entre l'appareil et un objet cible	[0]valeur
Temperature	1	Celsius	Mesure la température	[0]valeur



Exemple : Accéléromètre

```
public class MainActivity extends AppCompatActivity
implements SensorEventListener
{
private SensorManager mSensorManager;
private Sensor mSensor;
TextView txt;
```



Exemple : Accéléromètre

```
@Override  
protected void onCreate(Bundle savedInstanceState)  
{  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    mSensorManager=(SensorManager) getSystemService(SEN  
SOR_SERVICE);  
  
    txt= (TextView) findViewById(R.id.text);
```



Exemple : Accéléromètre

```
if(mSensorManager!=null)
{
    mSensor =
        mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
}
else
{
    Toast.makeText(this,"Sensor service not
detected",Toast.LENGTH_SHORT).show();
}
```



Exemple : Accéléromètre

```
@Override  
protected void onPause()  
{  
    super.onPause();  
    txt.setText("onpause");  
    mSensorManager.unregisterListener(this);  
}
```



Exemple : Accéléromètre

```
@Override  
protected void onResume() {  
    txt.setText("onresume");  
    super.onResume();  
    mSensorManager.registerListener(this,mSensor,mSensorMa  
    nager.SENSOR_DELAY_NORMAL);  
}
```



Exemple : Accéléromètre

```
@Override  
public void onSensorChanged(SensorEvent sensorEvent)  
{  
if(sensorEvent.sensor.getType()==Sensor.TYPE_ACCELEROMETER)  
{  
    int x=(int)sensorEvent.values[0];  
    int y=(int)sensorEvent.values[1];  
    int z=(int)sensorEvent.values[2];  
  
    txt.setText("x="+x+" y="+y+" z="+z);  
}  
}
```



Exemple : Accéléromètre

```
@Override  
public void onAccuracyChanged(Sensor sensor, int i) {  
}  
}
```



Utilisation du filtre Google Play pour configurer une application avec capteur

```
<uses-feature  
    android:name="  
        android.hardware.sensor.accelerometer"  
    android:required="true" (false)  
/>
```

<uses-feature>

Dans le manifest pour filtrer les appareils qui ont la configuration de capteurs requise par l'application

