

Smartphone data documentation

(Khalil.Hajji@sony.eu.com)

- First Level:
 - **activityRecognitionResult:** Sony activity recognition result. Gives information about the user like sitting, carrying ect...
 - **androidActivityRecognitionResult:** Google activity recognition result. Gives information like on_car, on_bicycle.
 - **app launch :** information about the application launched
 - **battery:** information about the battery of the phone
 - **bluetooth:** information about the Bluetooth connection
 - **event:** information about the event that occurred, especially the time zone
 - **notifications:** information about the notifications if any
 - **headsetPlug:** information about the headset current use (if any)
 - **launcherLayouts**
 - **location:** information about the geographic location of the user
 - **networkInfo:** information about the phone network (internet connection or Bluetooth connection..)
 - **predictors:** results of the different predictors on the launch of this application
 - **sensor:** values of the different sensors of the smartphone
 - **settingInfo:** the actual information about the phone settings (ringermode, roamingEnabling, ...)
 - **telephony:** information about the telephony carrier of the user and the station used
 - **wifiApps:** information about the apps that are connecting or using internet
 - **wifiConnectedAp:** information about some wifi and internet details of the connected application
- **activityRecognitionResult:**
 - **activity:** represents the activity that the user is doing. The values are:
 - ✗ The device is put apart from a body
 - ✗ Unrecognizable motion
 - ✗ Jumped
 - ✗ Going up in an elevator
 - ✗ Going down in an elevator
 - ✗ Walking
 - ✗ Running
 - ✗ Riding a bicycle
 - ✗ Sitting in a moving car

- ✗ Standing in a moving train
 - ✗ Sitting in a moving train
 - ✗ In a moving train
 - ✗ Standing in a moving bus
 - ✗ Sitting in a moving bus
 - ✗ In a moving bus
 - ✗ Standing still
 - ✗ Sitting still
 - ✗ still
- **carrying:** Indicates whether the user has it's phone or not. Values are:
 - ✗ unknown
 - ✗ carrying
 - ✗ not carrying
- **createDate:** the exact date and time of the creation in milliseconds (in UNIX).
- **motion:** indicates information about how the user is moving. The values are:
 - ✗ Unknown
 - ✗ Jump
 - ✗ Elev Up
 - ✗ Elev Down
 - ✗ Esc Up
 - ✗ Esc Down
 - ✗ Stairs Up
 - ✗ Stairs Down
 - ✗ Walk
 - ✗ Run
 - ✗ Bicycle
 - ✗ Car
 - ✗ Train
 - ✗ Bus
 - ✗ Still
- **posture:** indicates information about the posture of the user. The values are:
 - ✗ Unknown
 - ✗ Sit
 - ✗ Stand
- **screen:** indicates whether the screen is on or off. The values are:
 - ✗ Unknown
 - ✗ ScreenOff
 - ✗ ScreenOn
 - ✗ UserPresent
- **seq:** an integer that is incremented that represents the id of the whole record. That means that the seq number is the same for all the features of the same record.

- **usb:** indicates whether there is a usb plug in or not. The values are:
 - ✗ Unknown
 - ✗ Disconnected
 - ✗ Connected
 - **version**

- **androidActivityRecognitionResult:** See the class DetectedActivity of android for exhaustive information:

<https://developer.android.com/reference/com/google/android/gms/location/DetectedActivity.html>

 - **activities:** list of the activities that the user may do. Activities might be:
 - ✗ IN_VEHICLE
 - ✗ ON_BICYCLE
 - ✗ ON_FOOT
 - ✗ RUNNING
 - ✗ STILL
 - ✗ TILTING
 - ✗ UNKNOWN
 - ✗ WALKING
 - **activity:** the most probable activity of the user
 - **confidence:** the confidence (i.e the certitude) about the fact that the user is doing this activity. Confidence is a number between 0 (no_certitude) and 100 (complete certitude)
 - **confidences:** the list of the certitudes of the activities that the user may do
 - **createDate:** the exact date and time in milliseconds of the activity. The value represents the number of milliseconds that occurred since 1st January 1970 UTC (UNIX). There might be some seconds or difference between the time of the record of the activity and the app launch.
 - **seq:** an integer that is incremented that represents the id of the whole record. That means that the seq number is the same for all the features of the same record.
 - **time:**

- **app launch:**
 - **app name:** the name of the application launched
 - **argument:** it is documented as an additional argument for applunch. But in reality this argument is usually empty.
 - **createDate:** the exact date and time of the creation in milliseconds (in UNIX).
 - **seq:**

- **battery:** See class BatteryManager of android for exhaustive information:
<http://developer.android.com/reference/android/os/BatteryManager.html>
 - **createDate:** the exact date and time of the record of the battery in Unix format. It can differ from createDate of the app launch by few seconds.
 - **health:** an integer representing the health of the battery (based on the android sensor extraction, class BatteryManager).
 - ✕ 2 : Battery health is Good
 - ✕ 7 : Battery health is Cold
 - ✕ 4 : Battery health is Dead
 - ✕ 1: Battery health is Unknown
 - **icon:** integer containing the resource ID of a small status bar icon indicating the current battery state.
 - **level:** integer representing the current battery level. It goes from 0 to scale.
 - **plugged:** an integer representing the plugging system currently used by the user to charge the device.
 - ✕ 0: no plugger, consuming from battery
 - ✕ 1: AC charger
 - ✕ 2: USB port
 - ✕ 4: wireless plugging (ex: energy tables)
 - **Present:** boolean indicating whether a battery is present
 - **scale:** the maximum level that the battery can achieve.
 - **seq**
 - **status:** integer representing the status of the battery.
 - ✕ 2: battery status charging
 - ✕ 3: battery status discharging
 - ✕ 5: battery status is full
 - ✕ 4: battery status not charging
 - ✕ 1: battery status unknown
 - **technology:** String describing the technology of the current battery
 - **temperature:** integer containing the current battery temperature in degree Centigrade.
 - **Voltage:** integer containing the current battery voltage level in millivolts.
- **bluetooth[]:** each child entry represents information about one device (a Bluetooth device that is detected by the smartphone). The exhaustive information about a device can be found in the class BluetoothDevice of android :
<http://developer.android.com/reference/android/bluetooth/BluetoothDevice.html>
 - **address:** the mac address of the remote device
 - **bondState:** the bonding state of the current connection

- ✗ 12: BOND_BONDED, the remote device is bounded
 - ✗ 11: BOND_BONDING, bounding is in progress
 - ✗ 10: BOND_NONE, the remote device is not bounded
 - **createDate:** the time and date in UNIX of the preleved information (That is to say that for one Bluetooth entry, all the children trees should have the same value of createDate since this information was preleved at the same time).
 - **name:** the name of the remote Bluetooth
 - **seq**
 - **type:** the type of the remote device
 - ✗ 1: DEVICE_TYPE_CLASSIC
 - ✗ 3: DEVICE_TYPE_DUAL
 - ✗ 2: DEVICE_TYPE_UNKNOWN
- **event:**
 - **createDate:** the creation date of the event (UNIX)
 - **guid:** a unique id of the user
 - **seq:**
 - **time:**
 - **timeZone:** the time zone where the device is
 - **timeZoneOffset:** the difference in milliseconds between the time zone and utc
 - **type:** the type of event, example `applaunch_from_launcher_prediction`. The different types of the events are:
 - ✗ `notification`: notification received generated this event
 - ✗ `app_launch`: a launch of an application generated this event
 - ✗ `screen_on`: the screen of smartphone became on
 - ✗ `screen_off`: the screen of the smartphone became off
 - ✗ `launcher_on`
 - ✗ `launcher_off`
 - ✗ `duplicated_notification`: indicates that the event was generated due to a duplicated notification. Note that this value is present only in the filtered notification version of the data.
- **notifications[]:** see the class Notification of Android for exhaustive information (<http://developer.android.com/reference/android/app/Notification.html>). Each time a notification is recorded it is added to the array of notifications. When a notification is opened or cleared it disappears from the array. Note that there is some notifications recorded that comes from the system and are not seen by the user.
 - **createDate:** the time and data in UNIX of the publishing or the notification. It is not the time when the notification occurred but the time when the record was outputted.

- **isClearable:** indicates whether the notification can be cleared by the user or not. Takes 1 if clearable, 0 otherwise.
 - **isOnGoing:** indicates whether the notification makes reference to an ongoing event (ex: current phone call). Takes 1 if the event is ongoing, 0 otherwise.
 - **packageName:** the name of the application that issued the notification.
 - **postTime:** the time and date in UNIX that represents when the notification occurred.
 - **priority:** an integer parameter that indicates the priority of the notification.
 - ✕ -2 : PRIORITY_MIN
 - ✕ -1: PRIORITY_LOW
 - ✕ 0: PRIORITY_DEFAULT
 - ✕ 1: PRIORITY_HIGH
 - ✕ 2: PRIORITY_MAX
 - **seq:**
 - **sortIndex:** a sort key that orders this notification among other notifications from the same package. This can be useful if an external sort was already applied and an app would like to preserve this. Notifications will be sorted lexicographically using this value, although providing different priorities in addition to providing sort key may cause this value to be ignored.
- **headsetPlug:** see AudioManager class of android for exhaustive information:
http://developer.android.com/reference/android/media/AudioManager.html#ACTION_HEADSET_PLUG
 - **createDate:** the UNIX date corresponding to the prelevement of the headsetPlug information
 - **microphone:**
 - ✕ 1: headset has a microphone
 - ✕ 0: headset doesn't have a microphone
 - **name:** the type of the headset
 - **seq:**
 - **state:**
 - ✕ 0: headset not plugged in the phone
 - ✕ 1: headset plugged
 - **launcherLayouts[]:** represents the positions of the different applications on the screen
 - **appName**
 - **cellX**

- **cellY**
 - **createDate**
 - **directoryId**
 - **fixedAreaId**
 - **isWidget**
 - **screenId**
 - **seq**
-
- **location:** see the class Location of android for exhaustive documentation.
<http://developer.android.com/reference/android/location/Location.html>
 - **accuracy:** Get the estimated accuracy of this location, in meters. We define accuracy as the radius of 68% confidence. In other words, if you draw a circle centered at this location's latitude and longitude, and with a radius equal to the accuracy, then there is a 68% probability that the true location is inside the circle. In statistical terms, it is assumed that location errors are random with a normal distribution, so the 68% confidence circle represents one standard deviation. Note that in practice, location errors do not always follow such a simple distribution. This accuracy estimation is only concerned with horizontal accuracy, and does not indicate the accuracy of bearing, velocity or altitude if those are included in this Location. If this location does not have an accuracy, then 0.0 is returned.
 - **altitude:** the altitude if available, in meters above the WGS 84 reference ellipsoid. If this location does not have an altitude then 0.0 is returned.
 - **bearing:** the horizontal direction of travel of this device, and is not related to the device orientation. It is guaranteed to be in the range (0.0, 360.0] if the device has a bearing. If this location does not have a bearing then 0.0 is returned.
 - **createDate:** the date in UNIX of the creation of this location
 - **elapsedRealtimeNanos:** the time of this fix in nanoseconds. A time of fix is the time needed for a GPS receiver to acquire satellite signals and navigation data.
 - **latitude:** latitude in degrees
 - **longitude:** longitude in degrees
 - **provider:** the name of the provider that generated this fix
 - **seq:**
 - **speed:** the speed of the device over the ground in meters/seconds. 0.0 is returned if the location doesn't have a speed.
 - **time:** the the date in UNIX of the creation of this location

- **networkInfo:** It gives information about the internet connection. see the class `NetworkInfo` of android for exhaustive information:
<http://developer.android.com/reference/android/net/NetworkInfo.html>
 - **available:** indicates whether network connectivity is possible. 1 if available, 0 otherwise.
 - **connected:** indicates whether the network connectivity exists and it is possible to establish connections and pass data. 1 if connected, 0 otherwise.
 - **createDate:** the date and time in UNIX of the creation of this `NetworkInfo`
 - **detailState:** get the detailed state of the network. For exhaustive list, see `NetworkInfo.DetailedState`
<http://developer.android.com/reference/android/net/NetworkInfo.DetailedState.html>). Some values are
 - ✕ AUTHENTICATING
 - ✕ BLOCKED
 - ✕ CONNECTING
 - **extraInfo:** Report the extra information about the network state, if any was provided by the lower networking layers.
 - **reason:** Report the reason an attempt to establish connectivity failed, if one is available.
 - **roaming:** indicates whether the device is currently roaming on this network. 1 if the device is roaming, 0 otherwise.
 - **seq:**
 - **state:** the state of the network (see <http://developer.android.com/reference/android/net/NetworkInfo.State.html>)
 - ✕ CONNECTED
 - ✕ CONNECTING
 - ✕ DISCONNECTING
 - ✕ SUSPENDED
 - ✕ UNKNOWN
 - **subType:** an integer representing the subtype of the network (the exhaustive list can be found in `TelephonyManager.NETWORK_TYPE_*` (see: <http://developer.android.com/reference/android/telephony/TelephonyManager.html>). Some values are
 - ✕ 0: `NETWORK_TYPE_UNKNOWN`
 - ✕ 4: `NETWORK_TYPE_CDMA`
 - **subTypeName:** the readable human version of the network subtype. Ex CDMA.
 - **type:** the type of network to which the device is connected. The exhaustive list can be found in `ConnectivityManager.TYPE_*` (see: <http://developer.android.com/reference/android/net/ConnectivityManager.html>). Some values are
 - ✕ 7: `TYPE_BLUETOOTH`
 - ✕ 8: `TYPE_DUMMY`

- ✕ 0: TYPE_MOBILE
 - ✕ 1: TYPE_WIFI
 - **typeName:** the human readable version of the network type.
- **Predictors[]:** The applications that were selected by the predictor. Each entry represents an application.
 - **appName:** the name of the predicted application
 - **createDate:** the date of the creation of the record
 - **itemOrder:**
 - **predictor:** the name of the predictor used
 - **score:** the prediction score
 - **selected:** 1 if this app is selected by the user, 0 otherwise
 - **seq:**
- **sensor[16]:** for complete information see Sensor class of android.
http://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ORIENTATION
 - **sensor[0]**
 - ✕ **createDate:** time and date in UNIX of the creation of the record
 - ✕ **seq**
 - ✕ **type:** 3 , corresponds to the **TYPE_ORIENTATION** sensor
 - ✕ **value[3] :** for complete information see SensorEvent class.
<http://developer.android.com/reference/android/hardware/SensorEvent.html>
 - Azimuth, angle between the magnetic north direction and the y-axis, around the z-axis (0 to 359). 0=North, 90=East, 180=South, 270=West
 - Pitch, rotation around x-axis (-180 to 180), with positive values when the z-axis moves toward the y-axis.
 - Roll, rotation around the x-axis (-90 to 90) increasing as the device moves clockwise.
 - **sensor[1]:**
 - ✕ **createDate**
 - ✕ **seq**
 - ✕ **type:** TYPE_ACCELEROMETER
 - ✕ **value[3] :**
 - Acceleration minus Gx on the x-axis
 - Acceleration minus Gy on the y-axis.
 - Acceleration minus Gz on the z-axis.

- sensor[2]
 - ✗ ...
 - ✗ ...
 - ✗ **type: TYPE_GAME_ROTATION_VECTOR** The rotation vector represents the orientation of the device as a combination of an *angle* and an *axis*, in which the device has rotated through an angle θ around an axis $\langle x, y, z \rangle$.
 - ✗ ...
- sensor[3]
 - ✗ **type: TYPE_GEOMAGNETIC_ROTATION_VECTOR**
- sensor[4]
 - ✗ **type: TYPE_GRAVITY**
- sensor[5]
 - ✗ **type: TYPE_GYROSCOPE** All values are in radians/second and measure the rate of rotation around the device's local X, Y and Z axis. The coordinate system is the same as is used for the acceleration sensor. Rotation is positive in the counter-clockwise direction. That is, an observer looking from some positive location on the x, y or z axis at a device positioned on the origin would report positive rotation if the device appeared to be rotating counter clockwise.
- sensor[6]
 - ✗ **type: TYPE_GYROSCOPE_UNCALIBRATED**
- sensor[7]
 - ✗ **type: TYPE_LIGHT**
 - ✗ **value[3]**
 - Ambient light level in SI lux units
 - void
 - void
- sensor[8]
 - ✗ **type: TYPE_LINEAR_ACCELERATION** A three dimensional vector indicating acceleration along each device axis, not including gravity. All values have units of m/s^2 . The coordinate system is the same as is used by the acceleration sensor.
 - ✗ **value[3]**
- sensor[9]
 - ✗ **type: TYPE_MAGNETIC_FIELD** All values are in micro-Tesla (μT) and measure the ambient magnetic field in the X, Y and Z axis.
- sensor[10]
 - ✗ **type: TYPE_MAGNETIC_FIELD_UNCALIBRATED**
- sensor[11]
 - ✗ **type: TYPE_PRESSURE**
 - ✗ **value[3]**

- Atmospheric pressure in hPa (millibar)
 - void
 - void
 - sensor[12]
 - ✖ **type: TYPE_PROXIMITY**
 - ✖ **value[3]**
 - Proximity sensor distance measured in centimeters
 - void
 - void
 - sensor[13]
 - ✖ **type: TYPE_ROTATION_VECTOR**
 - sensor[14]
 - ✖ **type: TYPE_STEP_COUNTER** (Sony sensor)
 - sensor[15]
 - ✖ **type: TYPE_STEP_DETECTOR** (Sony sensor)
- **settingInfo[82]:** all the settings entry information of the phone
 - **telephony**
 - **cdmaCellLocBaseStationId:** the id of the base station used by the device
 - **cdmaCellLocBaseStationLat:** the latitude of the base station
 - **cdmaCellLocBaseStationLng:** the longitude of the base station
 - **cdmaCellLocNetworkId:**
 - **cdmaCellLocSystemId:**
 - **createDate:** the time and date in UNIX of the creation of the record
 - **gsmCellLocCid:**
 - **gsmCellLocLac:**
 - **gsmCellLocPrc:**
 - **networkCountryIso:**
 - **networkOperator:**
 - **networkOperatorName:**
 - **networkRoming:** indicates whether the phone is using roaming or not. 1 if the phone is using roaming, 0 otherwise.
 - **networkType:**
 - **networkTypeName:**
 - **phoneType:**
 - **phoneTypeName:**
 - **seq:**
 - **subscriberId:**

- **wifiAps[]**: The wifi-terminals that are detected by the device. Each entry represents a wifi terminal.
 - **bssid**: the mac address of the terminal
 - **capabilities**: the wifi proprieties of the terminal (WPA, WPA2...)
 - **createDate**: the date and time in seconds in UNIX of the record
 - **frequency**:
 - **level**:
 - **ssid**:
 - **timestamp**: the date and time in UNIX that corresponds to the moment where the terminal was detected.
- **wifiConnectedAp**: The terminal on which the device is currently connected.
 - **bssid**: the mac address of the terminal
 - **createDate**: the date and time in seconds (UNIX) of the record
 - **level**:
 - **linkSpeed**: the link speed of the network
 - **networkId**:
 - **rsi**
 - **seq**
 - **ssid**