# Android MotionDnaSDK API Documentation

## Construction

### MotionDnaSDK(Context context, MotionDnaSDKListener motionDnaSDKListener)

Creates an instance of the MotionDnaSDK. This method requires an application context and a object that has conformed to the MotionDnaSDKListener interface (see callback section below for details)

## Permissions

### String[] getRequiredPermissions()

Returns a String array of the permissions needed to run MotionDnaSDK estimation. Use in conjunction with requestPermissions() on the applications activity.

## Control

### public void start(String developerKey)

Starts up the SDK with valid developer key as input. As no configuration is provided in this call, the SDK uses the defaults specified in the configuration table below. A successful or failed authentication will be reported through the reportStatus() callback.

### public void start(String developerKey, HashMap<String, Object> configurations)

Starts up the SDK with valid developer key as input. A dictionary with options from the Configuration table below determine how the SDK is configured on startup. Any options not changed in the dicitonary will maintain their defaults as specified in the table. A successful or failed authentication will be reported through the reportStatus() callback.

### public void stop()

Stops the execution of the SDK and frees up memory and any saved estimation state

### public void reset()

Equivalent to a stop() followed by a start() while preserving the developer key and any confguration from the initial start() call.

### public void pause()

Halts position estimation and collection of sensor data.

### public void resume()

Resumes position estimation and collection of sensor data.

### public void startForegroundService()

Allow the MotionDnaSDK to run while your app is backgrounded by adding a Foreground service to the app. Using this method will place a permenent notification in the tray while the app runs.

## Configuration

|  |  |  |  |
| --- | --- | --- | --- |
| **Keyword** | **Options** | **Default** | **Description** |
| model | String (“simple”, “standard”, “headmount”) | “standard” | Choose the type of model to use for motion estimation |
| gps | boolean | true | Use GPS to set an initial position and perform corrections when outdoors |
| corrected\_trajectory | boolean | false | Show path leading up to global position fix and a history the updated path then corrections happen |
| callback | double | 40ms | Rate (ms) at which estimation is delivered to app layer |
| motion\_source | String (“device”,”external”) | “device” | Determines the source of inertial sensor data used for position estimation |
| logging | boolean | false | Record log file for debugging with Navisens team |

## Change Global Estimation

### public void setGlobalPosition(double latitude, double longitude)

Manually assigns the user’s position in the global frame.

### public void setGlobalPositionAndHeading(double latitude, double longitude, double heading)

Manually assigns the user’s position and heading in the global frame.

### public void setGlobalHeading(double heading)

Manually assigns the user’s heading in the global frame. Units are in degrees.

## Change Cartesian Estimation

### public void setCartesianHeading(double heading)

Manually assigns the user’s heading in cartesian space. Parameter units are in degrees and must be contrained to the -180 to +180 frame.

### public void setCartesianPosition( double x, double y)

Manually assigns the user’s cartesian position to the postition (x,y) in cartesian space. Parameter units are in meters.

## External Sensors

### public void inputMotion(double timestamp, double roll, double pitch, double yaw, double accX, double accY, double accZ)

This method must be called at regular intervals with sensor data if you have started the SDK with configuration `motion\_source` set to `external`.

**Parameters**

Timestamp units must be in seconds. Accelerometer x, y, and z should be in G's and Gyro roll, pitch, and yaw unit should be in radians/sec.

### public void inputMotion(double timestamp, double roll, double pitch, double yaw, double accX, double accY, double accZ, double heading, double headingAccuracy)

This method must be called at regular intervals with sensor data if you have started the SDK with configuration `motion\_source` set to `external`.

**Parameters**

Timestamp units must be in seconds. Accelerometer x, y, and z should be in G's and Gyro roll, pitch, and yaw unit should be in radians/sec. Heading and heading accuracy must be in radians with heading expressed in a 0-2π frame increasing in a clockwise direction.

## SDK Info

### public static String SDKBuild()

This static method will return a string representing the current build number (e.g. NAVISENS-R20.08.17.15.56-MASTER).

### public static String SDKVersion()

This static method will provide the currention version number of the SDK (e.g. v2.0.0).

## Callbacks (MotionDnaSDKListener implementation)

### public void receiveMotionDna(MotionDna motionDna)

Provides a MotionDna object with the current position information and other characteristics of a users motion and status. See Estimation Properties below for further details.

### public void reportStatus(MotionDnaSDK.Status status, String message)

Reports to the developer any changes to the SDK or errors that are noteworthy. Status types are accompanied by a message providing additional details on the event that has occured and any action that should be taken.

## Estimation Properties

These are the values representing the estimation provided

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Variable/Getter** | **Type** | **Units** |
| **MotionDna** | getAttitude() | Attitude |  |
|  | getLocation() | Location |  |
|  | getClassifiers() | HashMap [String:Classifier] |  |
|  | getTimestamp() | double |  |
|  | getPath() | ArrayList<MotionDna> |  |
|  |  |  |  |
| **Attitude** | euler | Euler |  |
|  | quaternion | Quaternion |  |
|  |  |  |  |
| **Euler** | roll | double |  |
|  | pitch | double |  |
|  | yaw | double |  |
|  |  |  |  |
| **Quaternion** | x | double |  |
|  | y | double |  |
|  | z | double |  |
|  | w | double |  |
|  |  |  |  |
| **Location** | cartesian | CartesianLocation |  |
|  | global | GlobalLocation |  |
|  |  |  |  |
| **CartesianLocation** | x | double | meters |
|  | y | double | meters |
|  | z | double | meters |
|  | heading | double | degrees |
|  |  |  |  |
| **GlobalLocation** | latitude | double | WGS84 |
|  | longitude | double | WGS84 |
|  | altitude | double | meters |
|  | accuracy | enum GlobalLocationAccuracy (LOW, HIGH) |  |
|  |  |  |  |
| **Classifier** | prediction.label | String |  |
|  | prediction.confidence | double |  |
|  | predictionStats | HashMap [String : PredictionStats] |  |
|  |  |  |  |
| **PredictionStats** | duration | double | seconds |
|  | distance | double | meters |
|  | percentage | double |  |

## Simultaneous Localization and Mapping (SLAM)

### public void recordObservation(int identifier, double uncertainty)

Input an identifier when visiting and revisiting a landmark in an indoor setting. The ID must be accompanied by an uncertainty value in meters indicating the maximum possible distance the user could currently be from the ID'd landmark. This will provide the SDK with additional information that can enable improved corrections to a users position.

## Logging

### public static String activeLogFilePath()

If logging is enabled this will return the full path of the nav file generated during the SDK operation.

### public static String activeLogFilename()

If logging is enabled this will return the simple filename for the nav file generated during the SDK operation.