Nervousnet Mobile API’s

How-To build an Android Native Axon App

- ## Overview

This document specifies a How-To step-by-step guide of integrating the nervousnet Mobile Library into your Android Project to use the Mobile API’s.

A Nervousnet Extension Android application is any Android App that uses the nervousnet Mobile API’s to access sensor and other related data from Nervousnet Mobile application.

- ## Prerequisites

Since nervousnet Mobile APIs is based on the Android Services and Android Interface Definition Language (AIDL) it would be good to read and understand how it works at:

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

https://developer.android.com/guide/components/aidl.html

For using the nervousnet mobile API’s requires calling a remote interface defined with AIDL

https://developer.android.com/guide/components/aidl.html#Calling

- ## Tools

  - Eclipse
  - Android ADT Plugin for Eclipse
  - Android Studio (not tested, but importing the project should work)

- ## JavaDoc

  https://github.com/nervousnet/nervousnet-android/tree/master/Documents/Technical/Android/APIs

- ## Source Code

1) Nervousnet Project

   https://github.com/nervousnet/nervousnet-android

2) **Nervousnet Library Project (nervousnetLIB)**

   https://github.com/nervousnet/nervousnet-android/tree/master/MobileClients/Android/nervousnetLIB

3) **Sample Extension Project (LightMeter, Accelerometer & Noisemeter)**

(Inside GitHub project -> Mobile Clients -> Android -> Sample Extensions -> *)

https://github.com/nervousnet/nervousnet-android/tree/master/MobileClients/Android/SampleExtensions/Lightmeter
https://github.com/nervousnet/nervousnet-android/tree/master/MobileClients/Android/SampleExtensions/Accelerometer
https://github.com/nervousnet/nervousnet-android/tree/master/MobileClients/Android/SampleExtensions/Noisemeter

• Steps

1) Include the ***nervousnetLIB*** project as a library project of your base project.
2) Declare an instance of the ***NervousnetRemote*** Interface

```
protected NervousnetRemote mService
```

3) Implement ***ServiceConnection***.

```
private ServiceConnection mServiceConnection;
    mServiceConnection = new ServiceConnection() {

    @Override
    public void onServiceDisconnected(ComponentName name) {
            mService = null;
            mServiceConnection = null;
            }

    @Override
    public void onServiceConnected(ComponentName name, IBinder service) {
            mService = NervousnetRemote.Stub.asInterface(service); //Step 5
            }
    };
```

4) Call Context.bindService(), passing in your ServiceConnection implementation.

```
bindService(it, mServiceConnection, 0);
```

5) In your implementation of onServiceConnected(), you will receive an IBinder instance (called service). Call YourInterfaceName.Stub.asInterface((IBinder)service) to cast the returned parameter to YourInterface type.

```
mService = NervousnetRemote.Stub.asInterface(service);
```

6) Call the methods that you defined on your interface. You should always trap DeadObjectException exceptions, which are thrown when the connection has broken; this will be the only exception thrown by remote methods.

```
LightReading lReading = mService.getLightReading();
```

Check JavaDoc for a complete list of APIs.. NervousnetRemote interface are current supported functions.

7) To disconnect, call Context.unbindService() with the instance of your interface.

```
unbindService(mServiceConnection);
```

- ## Google Play Store Links:

1) Nervousnet Mobile App

   https://play.google.com/store/apps/details?id=ch.ethz.coss.nervousnet.hub

2) **Nervousnet – Sample Native Axon App List**

   LightMeter: https://play.google.com/store/apps/details?id=ch.ethz.coss.nervousnet.extensions.lightmeter

   NoiseMeter: https://play.google.com/store/apps/details?id=ch.ethz.coss.nervousnet.extensions.noisemeter

   Accelometer: https://play.google.com/store/apps/details?id=ch.ethz.coss.nervousnet.extensions.