|  |  |  |  |
| --- | --- | --- | --- |
|  |  | |  |
| Nervousnet HUB API’s  How-To build an Android Extension App |
|  |  |  | |

|  |
| --- |
| **Overview** This document specifies a How-To step-by-step guide of integrating the nervousnet HUB Library into your Android Project to use the HUB API’s.  A Nervousnet Extension Android application is any Android App that uses the Nervousnet HUB API’s to access sensor and other related data from Nervousnet HUB application. **Prerequisites** Since nervousnet HUB 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 HUB 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/tree/master/Documents/Technical/Android/APIs> **Source Code**  1. Nervousnet Project   <https://github.com/nervousnet/nervousnet>   1. **Nervousnet Library Project (nervousnetLIB)**   <https://github.com/nervousnet/nervousnet/tree/master/MobileClients/Android/nervousnetLIB>   1. **Sample Extension Project (LightMeter)**   (Inside GitHub project -> Mobile Clients -> Android -> Sample Extensions -> **LightMeter**)  <https://github.com/nervousnet/nervousnet/tree/master/MobileClients/Android/SampleExtensions/Lightmeter> **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*   1. 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**  *}*  *};*   1. Call Context.bindService(), passing in your ServiceConnection implementation.   *bindService(it, mServiceConnection, 0);*   1. 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);*   1. 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.   1. To disconnect, call Context.unbindService() with the instance of your interface.   *unbindService(mServiceConnection);* |

# **Google Play Store Links:**

1. Nervousnet HUB

[**https://play.google.com/store/apps/details?id=ch.ethz.coss.nervousnet.hub**](https://play.google.com/store/apps/details?id=ch.ethz.coss.nervousnet.hub)

1. **Nervousnet – Sample Extension App List**

**LightMeter:** [**https://play.google.com/store/apps/details?id=ch.ethz.coss.nervousnet.extensions.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**](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**](https://play.google.com/store/apps/details?id=ch.ethz.coss.nervousnet.extensions)**.**