**KinometrixBT Application**

**Environment**

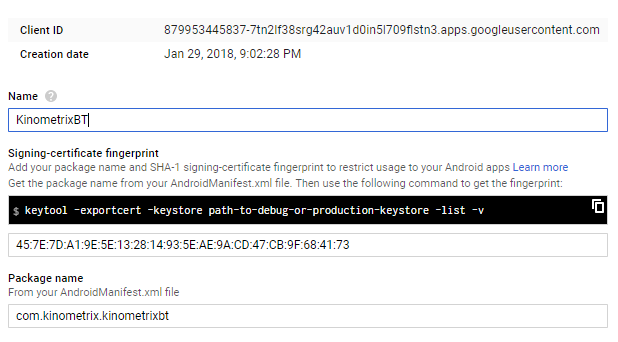
* Compile SDK version: 26
* Minimum SDK and Target Device Version: Android SDK 25, Android 7.1 Nougat
* Android Studio 3.0.1
* Windows 10 Pro

**Target Device specification**

|  |  |
| --- | --- |
| Type | Detail |
| CPU | Qalcomm SnapDragon 425 |
| Display | 8.0 Inch (Resolution 1280 pixels\*800 pixels) |
| Memory | LP DDR3 2GB |
| WLAN | 802.11 B/G/N |
| Bluetooth | BT 4.0 |
| Android Version | Android 7.1.1 |

**Software Library & Application Programing Interface**

* Google Drive API
  + Detail Information <https://developers.google.com/drive/>
  + Used API for GoogleDriveActivity.java
  + To use this API, log into the Google Developers console and create credentials for the usage, <https://console.developers.google.com/>
  + Here is my credential for development.



* + Add dependency

Add the following to the app level build.gradle:

|  |
| --- |
| build.gradle |
| // Google Drive Api  compile('com.google.api-client:google-api-client-android:1.23.0') {  exclude group: 'org.apache.httpcomponents'  }  // Google Play Service  compile group: 'com.google.apis', name: 'google-api-services-drive', version: 'v3-rev98-1.23.0'  compile 'com.google.android.gms:play-services-drive:11.8.0'  compile 'com.google.android.gms:play-services-auth:11.8.0' |

* MPAndroidChart
  + Graphic Chart Library
  + Free with Apache License 2.0

Unless required by applicable law or agreed to in writing, software distributed under the license is distributed on an “AS-IS” base, without warranties or conditions of any kind, either express or implied.

* + <https://github.com/PhilJay/MPAndroidChart>
  + Gradle Dependency

Add the following to the project level build.gradle:

|  |
| --- |
| build.gradle |
| allprojects {  repositories {  maven { url "https://jitpack.io" }  }  } |

* + Add this to the app level build.gradle:

|  |
| --- |
| build.gradle |
| implementation 'com.github.PhilJay:MPAndroidChart:v3.0.3' |

* ButterKnife
  + Annotate fields with @BindView and a view ID for Butter Knife to find and automatically cast the corresponding view in the layout.
  + <http://jakewharton.github.io/butterknife/>
  + Gradle Dependency

Add the following to the app level build.gradle:

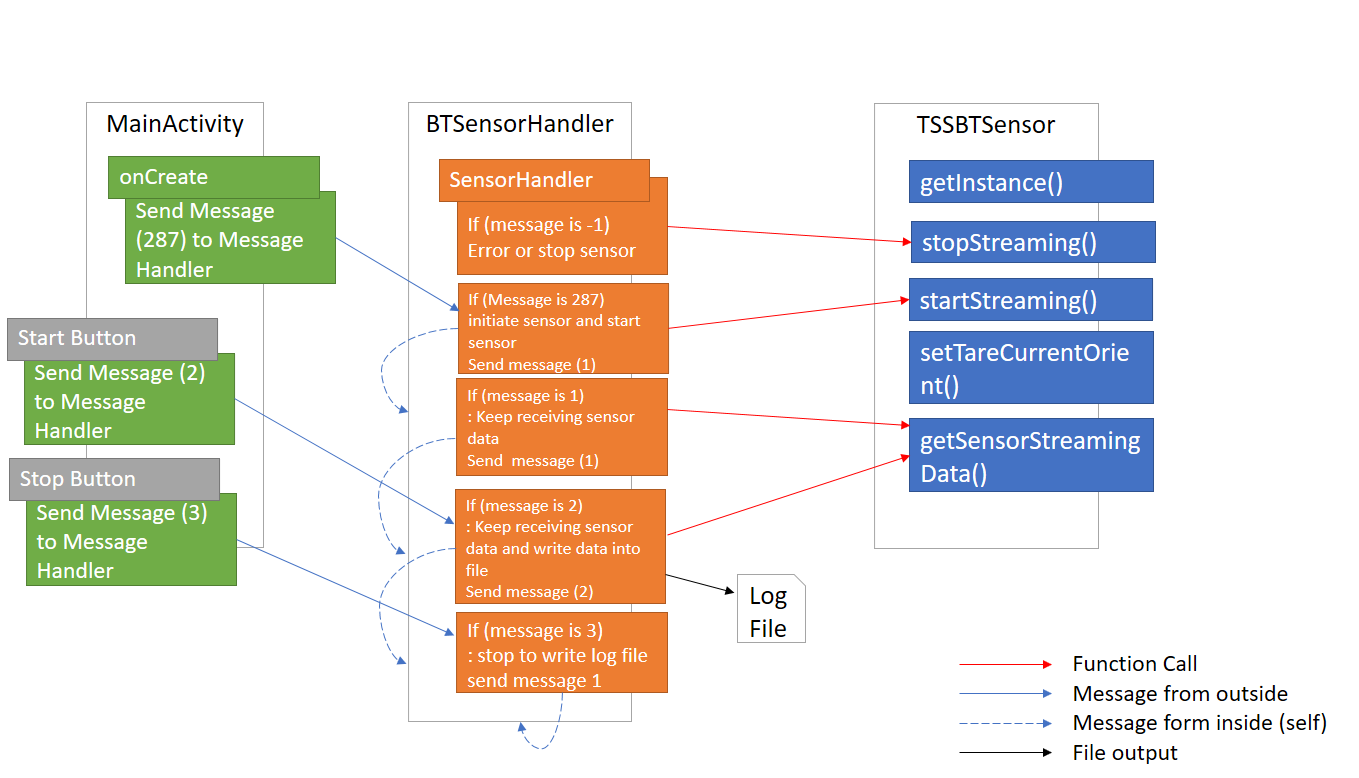
|  |
| --- |
| build.gradle |
| compile 'com.jakewharton:butterknife:8.5.1'  annotationProcessor 'com.jakewharton:butterknife-compiler:8.5.1' |

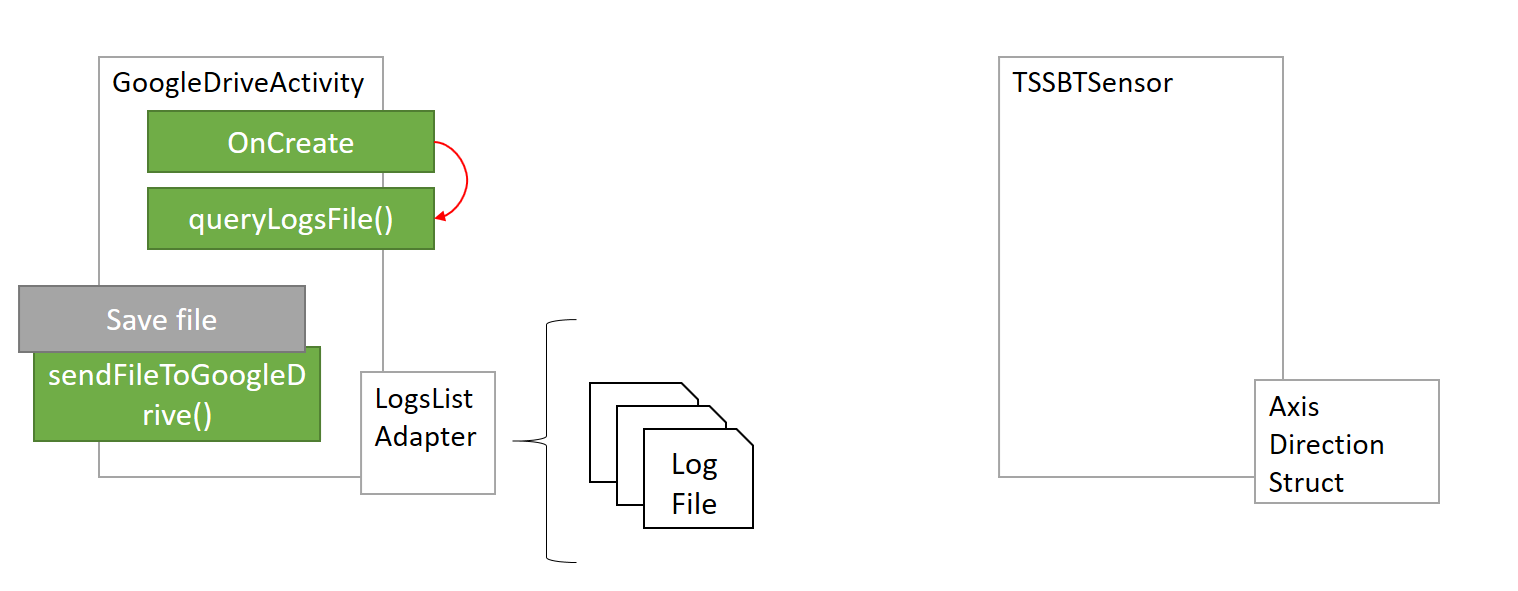
* EasyPermissions
  + EasyPermissions is a wrapper library to simplify basic system permission logic when targeting Android Marshmallow (sdk 23) or higher.
  + Gradle Dependency

Add the following to the app level build.gradle:

|  |
| --- |
| build.gradle |
| //EasyPermission  implementation 'pub.devrel:easypermissions:1.1.3' |

Overall Structure of the System





How to control the sensor device using SensorHandler

* Initiate the Bluetooth Sensor Handler

Add the following to the code

|  |
| --- |
| //Bluetooth Sensor Handler  private BTSensorHandler btSensorHandler = new BTSensorHandler(); |

* Initiate the sensor and start broadcasting sensor data

Add the following code to the code

|  |
| --- |
| Message start\_again\_message = new Message();  start\_again\_message.what = 287;  btSensorHandler.sendMessage(start\_again\_message);  is\_polling = true; |

\*Boolean type is\_polling indicates the current status of sensor. If it is true, which means the sensor is broadcasting its data otherwise broadcasting is stopped.

* Tare the sensor orientation

Use the following code to tare the sensor orientation

|  |
| --- |
| TSSBTSensor.getInstance().setTareCurrentOrient(); |

* Start logging function

Use the following code to create and start to save logs

|  |
| --- |
| Public static String logfile = “your file name”;  Message start\_exercise\_message = new Message();  start\_exercise\_message.what = 2;  btSensorHandler.removeCallbacksAndMessages(null); btSensorHandler.sendMessage(start\_exercise\_message); |

\*provide logfile name before calling btSensorHandler.sendMessage();

* Stop logging function

Use the following code to stop logging

|  |
| --- |
| Message stop\_exercise\_message = new Message();  stop\_exercise\_message.what = 3;  btSensorHandler.removeCallbacksAndMessages(null);  btSensorHandler.sendMessage(stop\_exercise\_message); |

* Getting sensor data

Use the following code to get sensor data

|  |
| --- |
| Float[] orient = TSSBTSensor.getInstance().getSensorStreamingData(); |
| // Quaternion  orient[0] : Sensor Quaternion X  orient[1] : Sensor Quaternion Y  orient[2] : Sensor Quaternion Z  orient[3] : Sensor Quaternion W  // Normal Gyrometer  orient[4] : Sensor Gyro X  orient[5] : Sensor Gyro Y  orient[6] : Sensor Gyro Z  // Euler Angle  orient[7] : Sensor Euler Angle Yaw  orient[8] : Sensor Euler Angle Picth  orient[9] : Sensor Euler Angle Roll  // Accelerometer  orient[10] : Sensor Accelerometer X  orient[11] : Sensor Accelerometer Y  orient[12] : Sensor Accelerometer Z |