

Build an Android App for IONO Analysis

Mohammed Khider
Software Engineer - Google

Advanced GnssLogger.apk

(advanced alternative to the Build-a-Simple-Logger tutorial that follows)

1. Download code (or github fork) from <https://github.com/google/gps-measurement-tools/tree/master/GNSSLogger>
 - The open-sourced App in Github has several advanced features to fork from.
2. Open build.gradle in Android Studio
3. Build, and try on your device
4. Make a change, and try it out

Example exercises:

5. Log file -
 - Change the logging output column order (e.g. to a format your organization's tools prefer)
6. Plots -
 - average signal strength of all observed SVs (or single strongest SV) instead of (or in addition to) top 4 SVs
7. AGNSS tab -
 - Create a new button for a delete-all test loop that finds the average, min & max TTFF over 10 tries

“Basic information on how to get started in the Android development ecosystem”

Getting started with apps

[Android Developers Channel](https://www.youtube.com/user/androiddevelopers)

<https://www.youtube.com/user/androiddevelopers>

[Official Android Developers Site](https://developer.android.com/index.html)

<https://developer.android.com/index.html>

[Making Your APP Location Aware](https://developer.android.com/training/location/index.html)

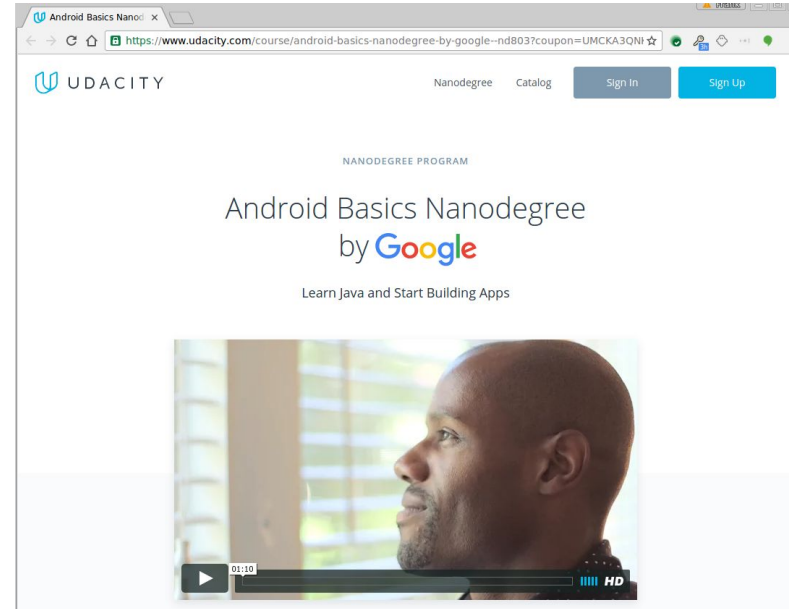
<https://developer.android.com/training/location/index.html>

[Location in Google Developer Channel](https://www.youtube.com/user/GoogleDevelopers/search?query=location)

<https://www.youtube.com/user/GoogleDevelopers/search?query=location>

[Location in Android Developer Channel](https://www.youtube.com/user/androiddevelopers/search?query=location)

<https://www.youtube.com/user/androiddevelopers/search?query=location>



← Back to Developers

DOWNLOAD

FEATURES

USER GUIDE

PREVIEW

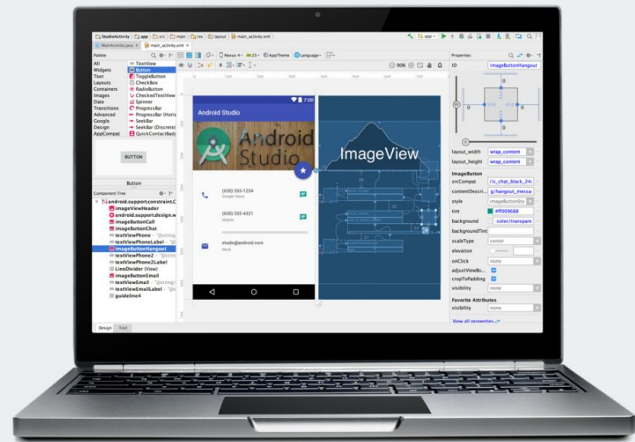
Android Studio

The Official IDE for Android

Android Studio provides the fastest tools for building apps on every type of Android device.

World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

DOWNLOAD ANDROID STUDIO
2.3.3 FOR MAC (463 MB)



<https://developer.android.com/studio/index.html>

Building your first Android APK

User interact with Android Apps using Activities. So each App should have a main Activity that get created when the App starts.

```
public class HelloWorldActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        TextView text = new TextView(this);
        text.setText("Hello World");
        setContentView(text);
    }
}
```

Tutorials on App development with Android Studio:

- <https://developer.android.com/training/index.html> Build simple App using Android Studio
- <https://developer.android.com/training/building-location.html> Build basic Location App
- http://www.tutorialspoint.com/android/android_hello_world_example.htm Hello World using

Android Studio

Build your own GnssLogger - Overview

User Interface

Main Activity

```

1 // MainActivity (onCreate) | new LocationListener (onLocation)
2
3 public class MainActivity extends AppCompatActivity {
4     private GpsLogger gpsLogger;
5     private LocationListener listener;
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11
12        gpsLogger = GpsLogger.getInstance();
13        listener = LocationListener.getInstance(getApplicationContext());
14
15        // Location UI
16        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
17        fab.setOnClickListener(new View.OnClickListener() {
18            @Override
19            public void onClick(View view) {
20                gpsLogger.startLogging();
21                listener.startListening();
22            }
23        });
24
25        // Location UI
26        FloatingActionButton fab2 = (FloatingActionButton) findViewById(R.id.fab2);
27        fab2.setOnClickListener(new View.OnClickListener() {
28            @Override
29            public void onClick(View view) {
30                gpsLogger.stopLogging();
31                listener.stopListening();
32            }
33        });
34
35        // New GPS measurements on button click
36        FloatingActionButton fab3 = (FloatingActionButton) findViewById(R.id.fab3);
37        fab3.setOnClickListener(new View.OnClickListener() {
38            @Override
39            public void onClick(View view) {
40                gpsLogger.sendFile();
41            }
42        });
43    }
44 }
                    
```

GnssController
Manage requests for location & measurements

UiLogger
Display incoming location & measurement messages on screen

FileLogger
Record measurements to file, and send file when done

LocationManager
(from Android System)

GnssListener
Common interface to listen for Location & Measurements

Build your own Gnss Logger (1/6)

- Download Android Studio? Then Create An Android Project

<https://developer.android.com/training/basics/firstapp/creating-project.html>

- Change the language to Java

- Run the HelloWorld App on Real Device or Emulator:

<https://developer.android.com/training/basics/firstapp/running-app.html>

- Setup the Layout Editor & Add Text Box and Button & Change UI String:

<https://developer.android.com/training/basics/firstapp/building-ui.html>

- Overview of Android Studio Layout
- Each activity has its xml layout that defines its appearance

Build your own Gnss Logger (2/6)

- Add the following Buttons to the Main Activity Screen with their respective IDs:
 - a. Name: Location ON, ID = locaiton_on
 - b. Name: Location OFF, ID = locaton_off
 - c. Name: Raw Meas. ON, ID = raw_meas_on
 - d. Name: Raw Meas. OFF, ID = raw_meas_off
 - e. Name: Start LOG, ID = start_log
 - f. Name: Send File, ID = send_file
- Add a text box below the buttons that our text will appear on it
- Ensure that the buttons are constraints to the parent layout and to each other (use Infer Constraints button and adjust if necessary)

Build your own Gnss Logger (3/6)

- Run the APK both on device and emulator and play with the buttons
- Get handles on all the buttons we added: [MainActivity-OnCreateSnippet1](#)
- Set the button to respond to click events: [MainActivity-OnCreateSnippet2](#)
- Create a class to handle location and raw measurements registration:
[GnssController-Snippet1](#)
- Add Location and Measurements callbacks to GnssController:
[GnssController-Snippet2](#). Lets re-run before going to next step.
- Build container around our Text Box for setting text appearance:
[MainActivity-Snippet3](#). Did we give Text View the right name in the Layout?

Build your own Gnss Logger (4/6)

- Add an interface with common methods to receive location and measurements: [GnssListener-Snippet](#)
- Build a UiLogger that listens to measurements and logs them to the screen (implements a GnssListener): [UiLogger-Snippet](#)
- Build a FileLogger that listens to measurements and logs them to file (implements a GnssListener): [FileLogger-Snippet](#)
- Fill code for register/unregister location and measurement methods in Main Activity: [MainActivity-OnCreateSnippet4](#)

Build your own Gnss Logger (5/6)

- In Main Activity, construct UiLogger and FileLogger and pass them GnssController for logging: [MainActivity-OnCreateSnippet5](#)
- Modify GnssController to log measurements to UiLogger and FileLogger: [GnssController-Snippet3](#)
- In Main Activity, fill code for Start Log and Send Log in the Main Activity: [MainActivity-OnCreateSnippet6](#)
- In Main Activity, request needed location runtime permissions: [MainActivity-Snippet6](#)

Build your own Gnss Logger (6/6)

- Edit Android Manifest to set required permissions and File Sharing options:
[AndroidManifest-Snippet](#)
- Create a resource folder **res/xml** and create in it a file named [provider_paths-Snippet](#). The file contains the path we want to log to/send files from. It is a good habit to put resources together!
- Build and Run in Emulator and Real Device

Build your own Gnss Logger - All the code

- Missed something during the previous sections? Cut-and-paste entire files:
 - Under app/java/com.example...
 - MainActivity.java
 - GnssController.java
 - GnssListener.java
 - FileLogger.java
 - UiLogger.java
 - Under app/res
 - Create directory 'xml'
 - Add file [provider_paths.xml](#)
 - Under app/manifests
 - Edit existing AndroidManifest.xml to set required permissions and File Sharing options:
[AndroidManifest-Snippet](#)

How far did you get?

- Do you have a working app?
 - You can use it to collect files for the same Matlab & GnssAnalysisApp tools we used this morning!

- Try it out during the final break & let us know as we wrap up.
- Full Code of all classes can be found [here](#)

End of Tutorial