**Localization API User Guide**

Modify History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Comment |
| 2014/6/11 | Lu Zhiyong | V1.0 | Finish key feature and getting started |
|  |  |  |  |

Index

[Key Developer Features 2](#_Toc390252221)

[1. Indoor Positioning 2](#_Toc390252222)

[2. Power Saving 2](#_Toc390252223)

[3. Offline Calculation 2](#_Toc390252224)

[Getting Started 2](#_Toc390252225)

[1. Apply API access permission 2](#_Toc390252226)

[2. Download the positioning data package 4](#_Toc390252227)

[3. Add Android permissions 4](#_Toc390252228)

[4. Using the library in your application 4](#_Toc390252229)

[5. Initialization an instance of IndoorLocationManager 5](#_Toc390252230)

[6. Setting the callbacks and start positioning 5](#_Toc390252231)

[7. Stop positioning 6](#_Toc390252232)

# Key Developer Features

## Indoor Positioning

The API provides the ability to get the user’s position in the indoor environment. You just need to open the WIFI permissions to it, and then it will automatically collect the WIFI information nearby and calculate the position that the user is in. The accuracy can be less than 5 meters.

## Power Saving

As we all know the WIFI on smartphones costs so much power. If one application keeps scanning the WIFI all the time, nobody will use it. The API can automatically saving the power when positioning. If the user’s movement is really slow, it will not collect WIFI information and update the user’s position.

## Offline Calculation

The whole positioning algorithm is offline, so you do not need to send all the data collected to the server. This means it can save the user’s 2G/3G/4G data traffic and save the user’s phone’s power.

# Getting Started

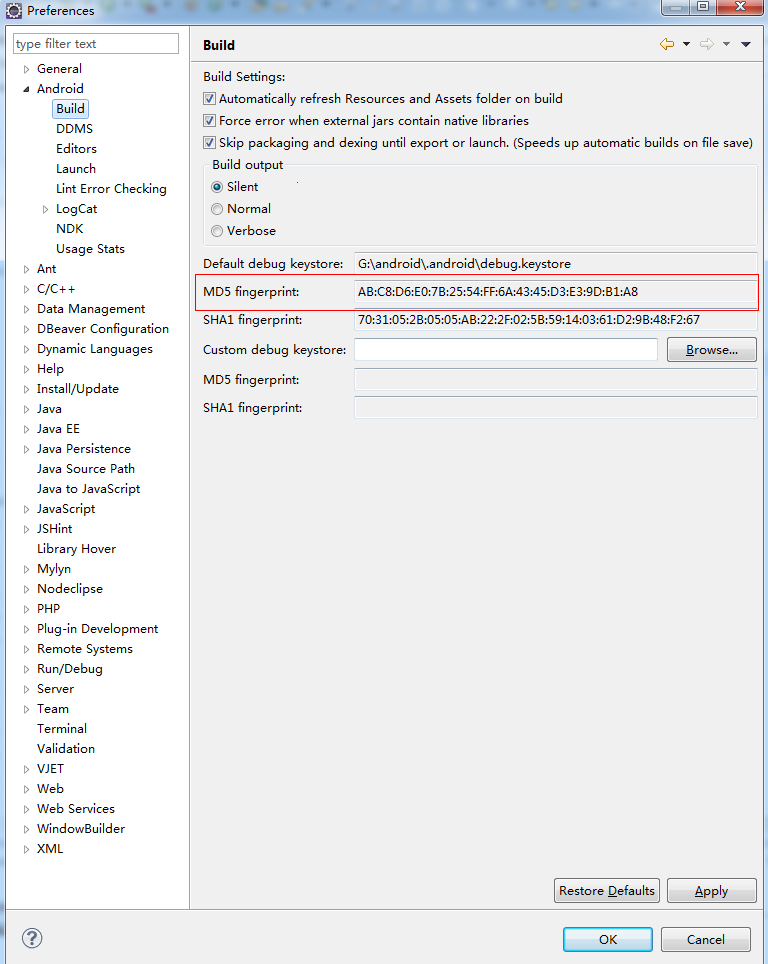
## Apply API access permission

Before using the Localization API, you need to access the website to apply for the permission about using the API. You must provide some information about you application to ask for the permission. You should provide the following information:

* Application name
* Application package name
* Application description
* MD5 fingerprint of your private key to pack your application
* Developer’s information

There is two way to get the MD5 fingerprint of the private key.

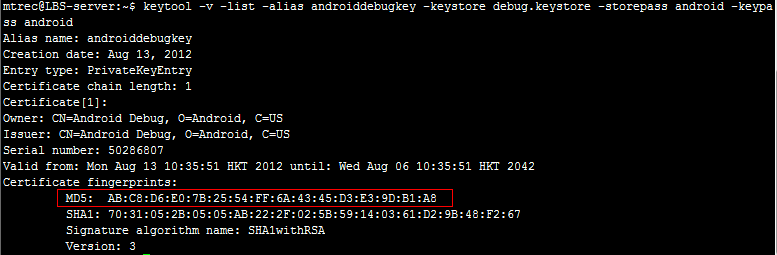
1. If you use the eclipse, you can go to **Window > Preference,** Open the **Android** tag and click the **Build** item, then you can find your fingerprints of the keystores. Just like the following image.



1. If you want to publish you application and using the keystore named “publish.keystore” to sign the application, you could run the follow command to get the MD5 fingerprint.

**<path to JDK/bin>**keytool -v -list -alias **<keyname>** -keystore **<publish.keystore>** -storepass **<keystore’s password>** -keypass **<key’s password>**

Replace the content inside the angle brackets with yours. And then you can get the information just like the following.



## Download the positioning data package

When your application is approved to use the Localization API, you can go back to our website and login to download the library and the positioning data package.

We provide some packages for downloading:

* **Localization API library** : it contains some .jar files and .so files to do the positioning work
* **Localization API document** : the documents that tell about how to use the API
* **Positioning Data** : the data that use to do the calculation
* **Map Data** : the maps of the site

For the positioning data, you can unzip it and put it to the SD card or the private path of your application, where it can be access by your application and the API.

## Add Android permissions

After creating your Android project, you need to add some permission to let the API work for you. You should add the permissions to the AndroidManifest.xml file.

The following permissions are required.

* android.permission.INTERNET
* android.permission.ACCESS\_WIFI\_STATE
* android.permission.CHANGE\_WIFI\_STATE

The following permissions are optional.

* android.permission.WRITE\_EXTERNAL\_STORAGE (If you put the positioning data on the SD card, this permission is needed)

## Using the library in your application

After download the library package, you can find some .jar files and some folders contain the .so file. These are the library files required for using the Localization API.

You should put all the files and folders into the directory named libs under the project’s root. And then add the jars to the build path.

## Initialization an instance of IndoorLocationManager

Before you get an instance of IndoorLocationManager, the API will first send a request to verify whether your application is approved to use the API. So you should use the following code to get an instance.

private IndoorLocationManager indoorLocationManager = null;

//…

IndoorLocationManager.newInstance(this, siteName, POSITION\_DATA\_PATH, new OnVerificationListener() {

public void onVerificationFinished(boolean isSuccess, String resultCode) {

if(isSuccess) {

//Verification passed!

indoorLocationManager = IndoorLocationManager.getInstance();

startPositioning();

} else {

//Verification failed

}

}

});

## Setting the callbacks and start positioning

The API uses an event-driven design. So you should setup some callback function to receive the information and result. The listeners you can set are list below:

* **OnWifiNotEnabledWhenScanning** : The API finds that the WIFI is closed when scanning
* **OnWifiConnected** : The callback when the phone connects to a WIFI
* **OnReadingDataBegin** : The callback tells you it starts reading data
* **OnGetScanResult** : The callback return you the scan result of the AP information
* **OnGetLocationResult** : The callback return the position after calculating

Here is the sample code.

private void startPositioning() {

indoorLocationManager.setOnWifiNotEnabledWhenScanningListener(new OnWifiNotEnabledWhenScanningListener() {

public void onWifiNotEnabledWhenScanning(WifiManager WifiManager) {

// TODO Handle the case that the WIFI is disabled when scanning

}

});

indoorLocationManager.setOnWifiConnectedListener(new OnWifiConnectedListener() {

public void onWifiConnected() {

// TODO Handle the case that the WIFI is connected

}

});

indoorLocationManager.setOnReadingDataBeginListener(new OnReadingDataBeginListener() {

public void onReadingDataBegin() {

// TODO The callback that the positioning process is pending to read data

}

});

indoorLocationManager.setOnGetLocationResultListener(new OnGetLocationResultListener() {

public void onGetLocationResult(String area, PointF[] points,

Integer[] radius, String symbol) {

// TODO The callback that finish calculating the position

if(area != null && points != null && points.length > 0) {

//TODO got the position

} else {

//TODO Calculate failed

}

}

});

indoorLocationManager.startLocation();

## Stop positioning

For stopping positioning, just call the following statements to stop it.

if(indoorLocationManager != null) {

indoorLocationManager.stopLocation();

}