Setting Up Gaia Android SDK vs 1.2.1

Add **GaiaSDK_v_121.aar** in libs folder of your main project.

• build.gradle

Set a flat filesystem directory as a repository for **GaiaSDK** aar included in the libs.

```
repositories{
   jcenter()
   flatDir{
      dirs 'libs'
   }
}
```

Include dependencies for Google Messaging Play Services and Google Location Services. A GCM play service is used to set cloud messaging android client and location play service is used for device location tracking.

Compile and include **GaiaSDK aar.**

```
dependencies {

  ...

  implementation 'com.google.android.gms:play-services-gcm:10.2.1
  implementation 'com.google.android.gms:play-services-location: 10.2.1'
  implementation 'com.google.android.gms:play-services-vision: 10.2.1'
  implementation 'com.amazonaws:aws-android-sdk-iot: 2.6.+'
  implementation 'android.arch.lifecycle:extensions: 1.1.0'
  implementation 'com.github.bumptech.glide:glide:4.6.1'
  implementation 'com.github.bumptech.glide:compiler:4.6.1'
  implementation (name:'GaiaSDK-1.2.1', ext:'aar')

}
```

• AndroidManifest.xml

Add Neoma Android name as **hk.neoma.gaiasdk.Neoma** as main application file name.

```
<application
    android:name="hk.neoma.gaiasdk.Neoma"
    ...

    >

    ...

</application>
```

• Setting Gaia SDK

Add the code to your Android Studio project in the Launching Activity to trigger and register the user with Gaia SDK and also to request for the required location permissions.

```
protected void onCreate(Bundle savedInstanceState) {

    // Set Neoma Api gateway key
      GaiaSDK.setNeomaApiGatewayKey(this, "<< NEOMAAPIGATEWAYKEY >>");

    // Set Neoma Client

    GaiaSDK.setClient(this, "<< CLIENT_ACCOUNT_KEY >>");


    GaiaSDK.setAdHandler(<<MainActivity.class>>);
}
```

Replace << **CLIENT_ACCOUNT_KEY** >> with your clientKey provided in the email. **setAdHandler** method initialises the activity that handles and open push notifications received from the server. Use **gaiaSDK.openAdSlider** method in this initialized activity.

• AdHandler Activity

Add the import statement for **GaiaSDK**.

```
import hk.neoma.gaiasdk.GaiaSDK;
```

In Activity set in setAdHandler function -

On onCreate method call the **openAdSlider** function of **GaiaSDK** to check if the push came and then clicked is done from the drop slider or when the application is in foreground.

If activity is a singleton **openAdSlider** should be called in onResume and onNewIntent also.

• User Registration And Permission Request

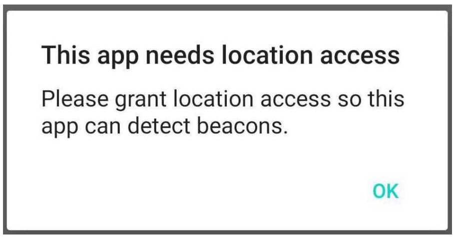
onRequestPermissionsResult is the callback when the necessary permissions like location and Bluetooth are granted by the user.

```
@Override
protected void onCreate(Bundle savedInstanceState) {

. . .
GaiaSDK.openAdSlider(this);
}
```

NOTE: Activity should be finished until the **onRequestPermissionsResult** callback is called after the **NeomaController.lookForPermission** method. **onRequestPermissionsResult** should be overridden within activity from which we are calling this method.

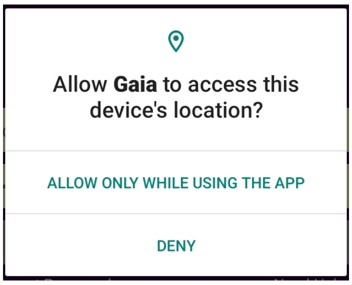
```
@Override
public void onRequestPermissionsResult(int requestCode, String permissions[],
int[] grantResults) {
    switch (requestCode) {
       case 2: {
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                Log.d("NEOMA", "Background location permission granted");
            } else {
                final AlertDialog.Builder builder = new
        AlertDialog.Builder(this);
                builder.setTitle("Functionality limited");
                builder.setMessage("Since location access has not been granted,
this app will not be able to discover beacons when in the background.");
                builder.setPositiveButton(android.R.string.ok, null);
                builder.setOnDismissListener(new
DialogInterface.OnDismissListener() {

                    @Override
                    public void onDismiss(DialogInterface dialog) {
                    }

                });
                builder.show();
            }
            return;
        }        case 1: {
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                Log.d("NEOMA", "coarse location permission granted");
                GaiaSDK.locationPermissionGranted();
            } else {
                final AlertDialog.Builder builder = new
AlertDialog.Builder(this);
                builder.setTitle("Functionality limited");
                builder.setMessage("Since location access has not been granted,
this app will not be able to discover beacons when in the background.");
                builder.setPositiveButton(android.R.string.ok, null);
                builder.setOnDismissListener(new
DialogInterface.OnDismissListener() {

                    @Override
                    public void onDismiss(DialogInterface dialog) {
                    }

                });
                builder.show();
            }
            return;
        }
    }
}
```

Basic Location request popup


This app needs location access

Please grant location access so this app can detect beacons.

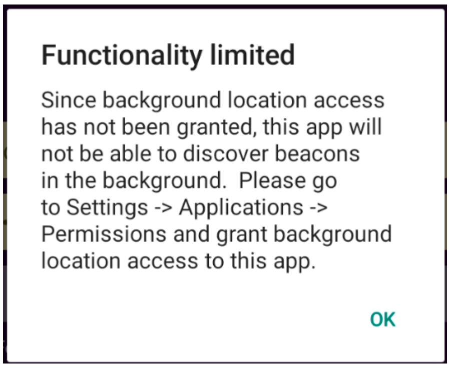
OK


Allow **Gaia** to access this device's location?

ALLOW ONLY WHILE USING THE APP

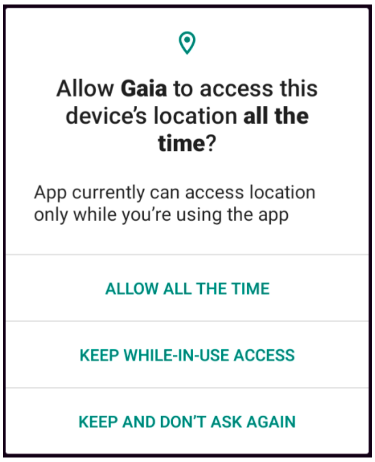
DENY

Location pop on AOS 29+ will ask for background permission separately


Functionality limited

Since background location access has not been granted, this app will not be able to discover beacons in the background. Please go to Settings -> Applications -> Permissions and grant background location access to this app.

OK


Allow **Gaia** to access this device's location **all the time**?

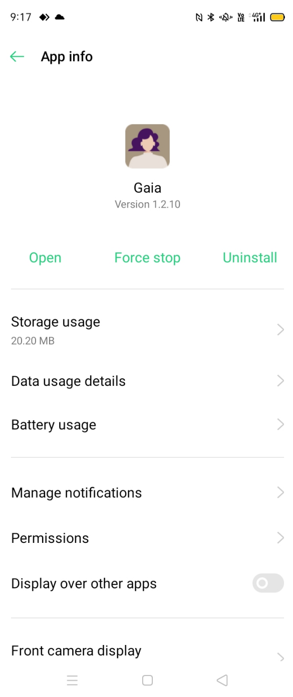
App currently can access location only while you're using the app

ALLOW ALL THE TIME

KEEP WHILE-IN-USE ACCESS

KEEP AND DON'T ASK AGAIN

Background permission pop may ask users to manual change permissions from settings.


9:17

App info

Gaia
Version 1.2.10

Open | Force stop | Uninstall

Storage usage
20.20 MB

Data usage details

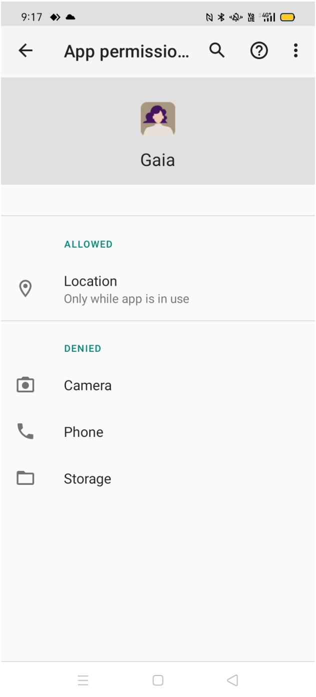
Battery usage

Manage notifications

Permissions

Display over other apps

Front camera display


9:17

App permissio...

Gaia

ALLOWED

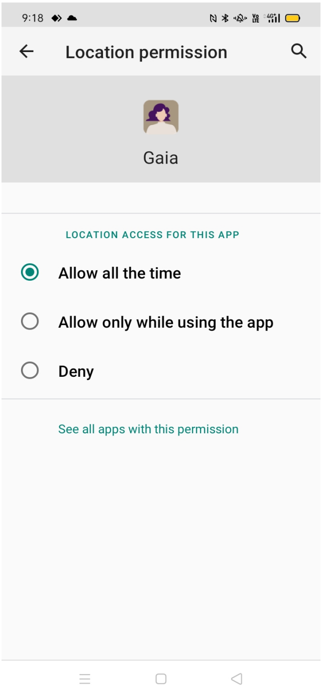
Location
Only while app is in use

DENIED

Camera

Phone

Storage


9:18

Location permission

Gaia

LOCATION ACCESS FOR THIS APP

● Allow all the time

○ Allow only while using the app

○ Deny

See all apps with this permission

• Saving User Profile

1. Call **saveProfile** method of **GaiaSDK** to save user's profile information from a JSONObject.

```
NeomaRequestManager
.saveProfile(<<JSONObject>>);
```

To **change** the single attribute in user profile, call **changeAttribute of GaiaSDK.** It will accept the attribute name and the value.

```
NeomaRequestManager
.changeAttribute(NeomaSDK.neomaAttributeNames.LANGUAGE,"chinese”);
OR
NeomaRequestManager
.changeAttribute(“nationality”,"chinese”);
```

App Usage

To track the how long a Activity is opened by a user in onResume function call **startTrackingActivity**. Please call stopTrackingActivity in onPause function. (Count is not monitored).

```
NeomaRequestManager
.startTrackingActivity(<<trackerName>>, true, <<ActivityClass>>);

NeomaRequestManager
.stopTrackingActivity();
```

To track the how long a Fragment is viewed by a user in function when view is being displayed on the activity, call **startTrackingFragment**. Call stopTrackingFragment when activity is closed or when no other fragment is getting tracked. (Count is not monitored).

```
NeomaRequestManager
.startTrackingFragment(<<trackerName>>, true, <<FramentClass>>);

NeomaRequestManager
.stopTrackingFragment();
```

To track the only how many times an Activity is opened (in onCreate funtion) or some button is clicked (in onClick function) or for any other action to be tracked. (Duration is not monitored). Please keep the tracker name same for duration and count when tracking activity or fragment

```
NeomaRequestManager
.startTracking(<<trackerName>>);
```

Display Notification List

You can display all saved notification list items by simply creating either a new Activity which extends `NotificationListActivity` class or a new Fragment which extends `NotificationListFrament` class of GaiaSDK respectively.

- ## Drawables resource

Replace your specific push Notification Image with this default Image with same name.

`ic_notification.png`

- ## Layouts resource

Add neoma_notification_top_header.xml in res/layout in the main project. Change Logo Image and Button Image of top header of Notifications as per your need without changing the Id's.

Add nm_list_item_notification.xml in res/layout in the main project to change the Default Layout of Notification History List as per your need.