

FLAMINGO

Implementation Guide

Have you received the Flamingo API already? If you have, awesome. You are one of the first developers to receive access to the world's first sub-metre positioning service for smartphones. We will explain in this leaflet the API's implementation and functionalities, so you can swiftly integrate it inside your project! Bear in mind, this is an alpha version of the service and there may still be some unencountered hiccups in the library. We appreciate all your feedback as we keep improving the service towards full operational status.

Step FLAMINGOAPI

Add `flamingoapi.aar` file inside your Android Studio project under the directory `app/libs`.

Step GRADLE

Add dependencies under `build.gradle(yourprojectmodule)`

1. Enable `.aar` libraries.

Change:

```
implementation fileTree(dir: 'libs', include: ['*.jar'])
```

to:

```
implementation fileTree(dir: 'libs', include: ['*.jar', '*.aar'])
```

2. Enable Java 8 features by adding:

```
compileOptions {  
    sourceCompatibility JavaVersion.VERSION_1_8  
    targetCompatibility JavaVersion.VERSION_1_8  
}
```

Source: <https://developer.android.com/studio/write/java8-support>

3. Add necessary third-party libraries:

```
implementation "com.google.guava:guava:28.1-jre"  
implementation group: 'com.squareup.okhttp3', name: 'okhttp', version: '3.12.0'  
implementation 'com.google.code.gson:gson:2.8.5'  
implementation 'joda-time:joda-time:2.9.4'
```

Step JAVA

Now you can use Flamingo features in you project! There are a few steps to receive Flamingo location inside your app.

A more detailed list of function definitions is provided at: <http://flamingosdk.com/android/>

1. PERMISSIONS Flamingo requires `ACCESS_FINE_LOCATION` and `READ_PHONE_STATE` permission as well as an Internet connection. <https://developer.android.com/training/permissions/requesting>
2. Create a `FlamingoManager` object and pass `Context` and a list of listeners.

```
FlamingoManager flamingoManager = new FlamingoManager(MainActivity.this, new ArrayList<>());
```

You can fill the list of listeners or add them separately (point 4).



3. Register Flamingo with your credentials.

```
flamingoManager.registerFlamingoService(applicationId, password, companyId);
```

Please ensure:

- Your credentials must be correct
- You need to be in the range of the reference station

You can ensure if the above conditions are fulfilled using:

```
flamingoManager.getReferenceStationStatus()
```

and

```
flamingoManager.getRegistrationStatus().
```

If the ReferenceStationStatus = AVAILABLE

and RegistrationStatus = FULL_ACCESS,

you are ready to enjoy the service!

4. Implement FlamingoLocationListener interface and its associated methods in your project.

```
public class MyClass extends AppCompatActivity implements FlamingoLocationListener {  
  
    @Override  
  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_my_class);  
        // create and register FlamingoManager  
  
    }  
  
    @Override  
    public void registerFlamingoLocationListener() {  
  
    }  
  
    @Override  
    public void unregisterFlamingoLocationListener() {  
  
    }  
  
    @Override  
    public void onFlamingoLocation(FlamingoLocation flamingoLocation) {  
  
        /* This method returns flamingoLocation object which contains flamingo coordinates.  
        For full list of methods check: http://flamingosdk.com/android/#flamingolocation */  
  
    }  
  
    @Override  
    public void onReferenceStationStatusChanged(ReferenceStationStatus referenceStationStatus) {  
  
        /* Returns the information if you are within the range of the reference station.  
        If your device is outside the zone covered by reference station the position will not be returned. */  
  
    }  
  
}
```

5. Override methods as you wish, for example:

```
@Override  
public void onFlamingoLocation(FlamingoLocation flamingoLocation) {  
    latitude = flamingoLocation.getLatitude();  
    longitude = flamingoLocation.getLongitude();  
}
```

6. Add the Flamingo listener:

```
flamingoManager.addFlamingoListener(flamingoListenerClass);
```

Happy coding !

