Android Public LOUIE7 SDK DOCUMENTATION



The Louie7 SDK collects user data according to permissions given by the user.

Integrate Louie7 Android SDK into your Android app

To integrate Louie7 SDK into your app, you have to follow these steps:

- 1. Download the archive. Unarchive it into the preferable folder. It can be something like /Users/user.name/Documents/repository/. The path to the *.aar file should be something like /Users/user.name/Documents/repository/com/louie7/louie-collectors/{current_sdk_version}. Make sure, that the first folder in the repository is com, not the louie-preview-sdk.
- 2. Add the following line in your build.gradle file.

```
repositories {
    google()
    mavenCentral()
    ...
    maven {
        url "path/to/your/repository"
    }
}
```

3. In your application level gradle file add the following code:

```
dependencies {
    ...
    implementation 'com.louie7:louie-collectors:{sdk_version}'
    ...
}
```

4. Sync your project.

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

Sync Now Ignore these changes

5. In your Application class in the onCreate method add the following code:

- 6. You successfully integrated Louie7 SDK into your app. Check Implement gesture collection guide to implement gesture collection.
- A Make sure, that you have added your application class in the Android manifest.
- 1 You can get your X-API-KEY from the dashboard.
- If you done everything right and there is internet connection on your test device, you will see the Connection with Louie7 servers established message.

Helpful information

If you are using proguard, make sure, that you have added these rules to the proguard-rules.pro:

```
-keep class com.louie7.** {*;}
-keepclassmembers enum * {*;}
```

The SDK works in background using coroutine worker. Only the phone's OS decides how much time background worker will be allowed to be active. Also it restarts only if the OS allows it. If the application stays in memory or goes in foreground, the SDK's worker will always start, so basically the SDK's lifetime closely connected both to application usage time/frequency and phone's memory and battery state. You can request ignore battery optimization, if it is possible.

Implement gesture collection

Gesture collection is strictly dependent on android views. Louie7 need access to views to collect gestures in your app. So, there is some options to implement gesture collection into your app.

Method 1: use GestureActivity as your parent activity.

- 1. In Android Studio select your main activity class.
- 2. Replace AppCompatActivity in your class with the GestureActivity.

```
class MainActivity : GestureActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
```

- 3. Repeat step 2 for every activity in your project.
- Make sure, that you have not missed any activity in your project.

Method 2: use GestureActivity as a parent class for your custom activity.

- 1. In Android Studio select your custom activity class.
- 2. Replace AppCompatActivity in your custom activity class with the GestureActivity.
- 3. Use your custom activity for every not abstract activity class in your project.

```
abstract class MyCustomActivity : GestureActivity() {
    fun internalFun1() {
        ///
    fun internalFun2() {
        ///
}
```

4. If you override dispatchTouchEvent method in your activity, make sure, that you have overridden it properly (both in custom and not abstract activities).

```
override fun dispatchTouchEvent(ev: MotionEvent?): Boolean {
        /// your code
        return super.dispatchTouchEvent(ev)
    }
```

Make sure, that you have not missed any activity in your project.

Method 3: use Louie7Gestures to implement gesture collection.

Use this method if you avoid unnecessary inheritance or can't use GestureActivity.

- 1. In Android Studio select your activity class.
- 2. Create a lateinit variable of Louie7Gestures.

```
private lateinit var gestures : GesturesLouieFamily
```

3. Initialize the GesturesLouieFamilyvariable with the Louie7Gestures.createListener function in the onCreate method of your activity.

```
override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        //your code
        gestures = Louie7Gestures.createListener(this)
    }
```

4. Override dispatchTouchEvent and call gestures.onTouchEvent(ev) (GesturesLouieFamily variable).

```
override fun dispatchTouchEvent(ev: MotionEvent?): Boolean {
        qestures.onTouchEvent(ev)
        return super.dispatchTouchEvent(ev)
    }
```

- 5. Your activity is ready to collect gestures now.
- If you want to use this method, you should repeat it for each activity you have.

How to integrate with SDK's API

To integrate your client application you have authorize your device identifier. To authorize your device identifier send a POST request to SDK's API.

All API requests require the use of a received application key. To authenticate an API request, you should provide your application key in the X-API-Key header.

| Header Parameter | Туре | Description |
|---------------------|--------|----------------------|
| X-API-Key | string | Your application key |

Next, to authorize a device identifier

You have send a POST request to SDK's API using received X-API-Key key.

Authorize device identifier

identifier

POST /api/v1/devices

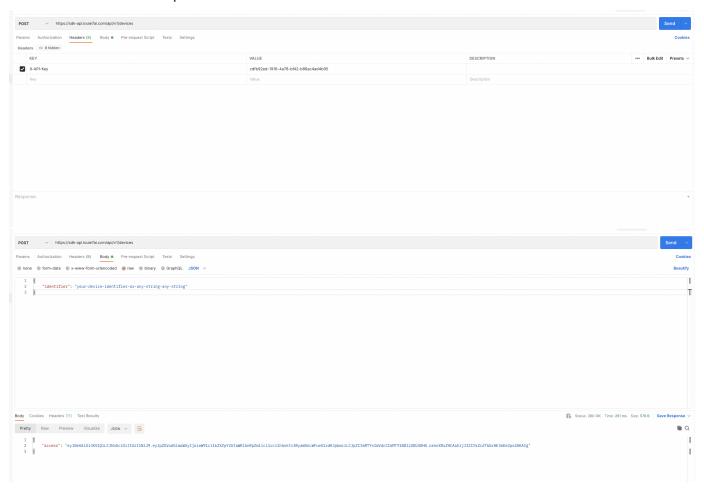
| Rody Parameter | Tyne | Description | |
|---------------------|------|-------------|--|
| | | | |
| 1001 / 451/ 11/ 401 | | | |

string

| Header Parameter | Туре | Description |
|------------------|--------|----------------------|
| X-API-Key | string | Your application key |

Device identifier

Authorize device identifier example



Now, that you've successfully authorized your application's device identifier, you request application's SDK's config.

Next, to receive application config

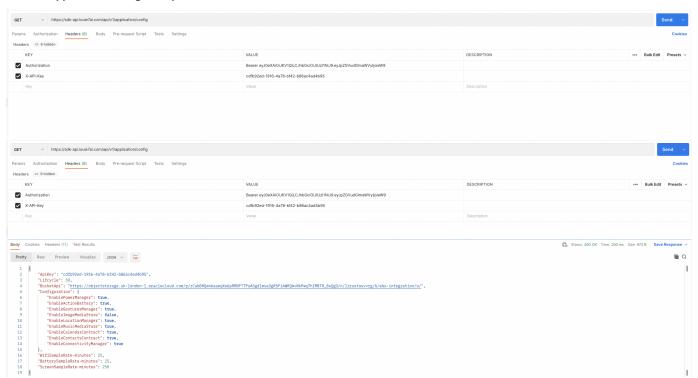
You have send a GET request to SDK's API using received access (Authorization) header.

Receive application config

GET /api/v1/application/config

| Header Parameter | Туре | Description |
|------------------|--------|-------------------------------|
| Authorization | string | Your device identifier access |

Receive application config example



You can also download the application config as a file

Download application config

GET /api/v1/application/config/download