



Android Mobile Risk SDK Integration Github (Kotlin)

Integration Overview

Integrations are done in a few simple steps:

1. Integrate Sardine's SDK into your application.
2. Set the Ingo provided `clientID` into the JS SDK. This ID is unique to the SDK and varies from your `participant_id`.
3. Call IngoPayments RiskSession API to obtain a `risk_session_token`.
4. Inject the `risk_session_token` as the `sessionKey` value inside the JS SDK.
5. Inject the `customer_ID` value that will be used in the subsequent RiskScore request as the `userIDHash` inside the JS SDK.
6. Call IngoPayments RiskScore API to obtain a risk level assessment of the transaction prior to committing.

Google Play Console update

Privacy Policy

Update your Google Play Store privacy policy with the following message: For fraud protection purposes only, (1) we collect your Battery Usage, Device Identifier, Device Storage, MAC Address, and > SIM information; (2) we also collect enough information to determine if you are trying to fake your current location by > > > using a VPN, VPN apps with location spoofing, or other related tools..

Data Safety

For the Data Safety section in developer console, please follow these > [steps](#) provided by Sardine. Note that you may need to add any other data types your app collects. The Developer Console provides an easier option to import the Data Safety information. You can import our Data Safety info > using this [CSV file](#) provided by Sardine.

Using Github Packages

1. Contact your Sardine team to provide you with your github access token.
2. Create `github.properties` file to your project root directory and add the following data:

```
gpr.usr=YOUR_GITHUB_USERNAME
gpr.key=YOUR_GITHUB_PERSONAL_ACCESS_TOKEN_WITH_read:packages_SCOPE
```

3. Add the following to project level `settings.gradle.kts` file:

```

val githubProperties = Properties().apply {
load(FileInputStream(rootDir.resolve("github.properties")))
}

dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
        maven {
            name = "GitHubPackages"
            url = uri("https://maven.pkg.github.com/sardine-sdk/SardineSDK-android")
            credentials {
                username = githubProperties['gpr.usr']
                password = githubProperties['gpr.key']
            }
        }
    }
}

```

4. Then add the following to the dependencies section in build.gradle.kts file:

```

dependencies{
    implementation "ai.sardine:sardinesdk:latest.release"

    // In case of using "Plus Telephony" version, use the following line instead
    //implementation "ai.sardine:sardinesdk-plus-telephony:latest.release"
}

```

Please double-check these permissions are in your AndroidManifest.xml file before proceeding further.

```

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.USE_BIOMETRIC" />
<uses-permission android:name="android.permission.USE_FINGERPRINT" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />

```

Note: Make sure to update your privacy policy whenever you add new permission.

If you use ProGuard, please keep `com.google.android.gms.`,
`com.google.android.gms.tasks.` ,
`com.google.android.gms.ads.identifier.AdvertisingIdClient.`

PERMISSION	USAGE
------------	-------

ACCESS_WIFI_STATE	Get User's Wifi name and Status
ACCESS_NETWORK_STATE	Get User's Network information and State
READ_PHONE_STATE	Get Phone and Network information (MNC, MCC, IMEI, Phone Number, Phone Type (GSM/CDMA), SIM Number, ...etc)
USE_BIOMETRIC USE_FINGERPRINT	Get user's Biometric authentication settings (Face or Fingerprint authentication)
WRITE_EXTERNAL_STORAGE	Write data into device to check re-installation behaviour
READ_EXTERNAL_STORAGE	Get External storage status, total size, free size...etc
ACCESS_COARSE_LOCATION ACCESS_FINE_LOCATION	Get user's GPS Location
READ_GSERVICES	Get Device GSF ID (Google Services Framework ID)

Initialize the SDK

Initialize Sardine SDK with sessionKey as soon as app starts. Ensure that you change the environment options to Production in Production/PlayStore apps.

```
Options option = new Options.Builder()
    .setClientID("YOUR_CLIENT_ID")
    .setSessionKey("SESSION_KEY")
    .setUserId("userID") // optional
    .setEnvironment(Options.ENV_SANDBOX) //Options.ENV_PRODUCTION
    .setSourcePlatform("Native")
    .build();
MobileIntelligence.init(context, option);
```