



What's New

Release Date	Aug 31, 2022
SDK Version	1.8.10
Summary	<p>Introduced capability</p> <ul style="list-style-type: none">• to connect BlockID wallet to DApps (decentralised application)• for transaction signing• for personal signing

Prerequisites

1. WalletConnect Project ID
 - a. Register a project into the cloud application
(<https://cloud.walletconnect.com/app>) to get Project ID with your project
 - b. The project ID allows you to start using the relay network immediately, and is needed during SDK initialization



Assumptions

This document provides detailed information about the dependencies, classes, methods, etc. introduced w.r.t. Wallet Connect Signing (Transaction and Personal) feature only.

Refer below SDK documentation for more details.

1. Overview:

<https://developer.1kosmos.com/devportal/docs/mobile-sdk/overview>

2. Integration Guide:

<https://developer.1kosmos.com/devportal/docs/mobile-sdk/android/integration-guide>

3. API Reference:

<https://developer.1kosmos.com/devportal/docs/mobile-sdk/android/api-reference>



Integration Changes

Android Studio

1. Android Studio Bumblebee (Android SDK 13) is used to build this version of the BlockID SDK.
2. It is recommended to use the same version for your application development.

Java Compatibility

The SDK is not binary compatible with applications that target lower than Java 11. In order to use this and future releases, you must upgrade applications to target Java 11.

```
compileOptions {  
    sourceCompatibility JavaVersion.VERSION_11  
    targetCompatibility JavaVersion.VERSION_11  
}
```

Proguard Rules

Add below proguard rules.

```
-keep class com.walletconnect.sign.**{* ;}  
-keep class net.sqlcipher.** { * ; }  
-keep class net.sqlcipher.database.** { * ; }
```



Gradle Dependencies - Project Level

Update below gradle build tool version.

```
classpath 'com.android.tools.build:gradle:7.2.1'
```

Gradle Dependencies - App Level

Add below configuration to resolve *bouncycastle* conflicts

```
configurations.all {  
    resolutionStrategy.eachDependency { DependencyResolveDetails details ->  
        if (details.requested.group == 'org.bouncycastle') {  
            details.useTarget 'org.bouncycastle:bcprov-jdk15to18:1.70'  
        }  
    }  
}
```

Update below dependencies in app level build.gradle file

```
// GSON  
implementation 'com.google.code.gson:gson:2.9.0'  
  
// Secure shared preference  
implementation 'androidx.security:security-crypto:1.1.0-alpha03'  
  
// Wallet  
implementation 'org.web3j:core:4.8.7-android'  
  
// Generate Mnemonics  
implementation('org.bitcoinj:bitcoinj-core:0.16.1')
```

// Annotation

implementation 'org.jetbrains:annotations:23.0.0'

// Base32 encode/decode

implementation 'commons-codec:commons-codec:1.11'

// Passport scan

```
implementation('org.jmrttd:jmrttd:0.7.32') {  
    exclude module: 'bcprov-jdk15on'  
    exclude module: 'bctls-jdk15on'  
}
```

// MFC scanner

implementation 'net.sf.scuba:scuba-sc-android:0.0.23'
implementation 'com.madgag.spongycastle:prov:1.58.0.0'

// CameraX

implementation 'androidx.camera:camera-camera2:1.2.0-alpha03'
implementation 'androidx.camera:camera-lifecycle:1.2.0-alpha03'
implementation 'androidx.camera:camera-view:1.2.0-alpha03'

// LiveID

implementation 'com.google.android.gms:play-services-mlkit-face-detection:17.0.1'

// Text recognition

implementation 'com.google.android.gms:play-services-mlkit-text-recognition:18.0.0'

// Barcode scanning

implementation 'com.google.mlkit:barcode-scanning:17.0.2'

// Opencv

```
implementation 'com.quickbirdstudios:opencv:4.5.3.0'
```

Add new dependencies in app level build.gradle file

```
// Wallet connect
```

```
implementation 'com.walletconnect:WalletConnectKotlinV2:2.0.0-rc.2'
```

```
// Web3
```

```
implementation("org.web3j:utils:5.0.0")
```

```
// Coroutines
```

```
implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.6.1'
```



Class Reference

WalletConnectHelper

A singleton class that provides applications to

- Connect BlockID SDK wallet to DApps
- Transaction signing
- Personal signing

Note: The SDK must be initialized and ready before using the WalletConnectHelper class.

Method Summary

public static WalletConnectHelper getInstance()

Description: Creates and returns a singleton instance of the WalletConnectHelper class.

public void initialize(@NonNull Application application, @NonNull String projectId, @NonNull Sign.Model.AppMetaData metadata, @NonNull WalletConnectCallback callback)

Description: Initializes the object based on given project id and application metadata.

public void connect(@NonNull String uri)

Description: Connect a client with a URI generated by the proposer.

public void approveConnection(@NonNull Sign.Model.SessionProposal

sessionProposal)

Description: Approves the requested session. A session is created between the proposer and responder.

public void rejectConnection(@NonNull Sign.Model.SessionProposal sessionProposal)

Description: Rejects the requested session.

public void approveSession(@NonNull Sign.Model.SessionRequest sessionRequest)

Description: Signs the transaction for a given session. Allowed methods are *eth_signTransaction* and *personal_sign* in the *SessionRequest* object.

public void rejectSession(@NonNull Sign.Model.SessionRequest sessionRequest)

Description: Rejects the transaction for a given session.

public List<Sign.Model.Session> getActiveSessions()

Description: Returns list of active sessions.

public void disconnect(@NonNull Sign.Model.Session session)

Description: Disconnect the given active session.

Interface Reference

WalletConnectCallback

Defines methods for callback events related to wallet connect/ disconnect, transaction and personal signing requests.

Method Summary

void onConnectionStateChange(Sign.Model.ConnectionState connectionState)

Description: A callback method to indicate the change in the proposer's connection state.

void onSessionProposal(Sign.Model.SessionProposal sessionProposal)

Description: A callback method to indicate that the proposer has sent a connection (pairing) request.

void onSessionRequest(Sign.Model.SessionRequest sessionRequest)

Description: A callback method triggered when a signing request (transaction, personal) is received.

void onSessionSettleResponse(Sign.Model.SettledSessionResponse.Result settleSessionResponse)

Description: A callback method indicating if there is change in the active session list.

void onSessionDisconnect(Sign.Model.DeletedSession deletedSession)

Description: A callback method triggered when the active session is disconnected.

void onError(Sign.Model.Error error)

Description: A callback method triggered when there is an error.