Cloudonix Mobile Application - Developer Guide

Tel: +972-73-2557799, Fax:+972-73-2557203, E-mail: info@cloudonix.io, P.O box 199, Udim, Israel 42905

Introduction

This is the mobile application developer guide for the Cloudonix Mobile SDK release 5.0. It will help you integrate the Cloudonix Mobile SDK into your Android or iOS mobile application.

Getting Started

The Cloudonix Mobile SDK is distributed with a sample dialer application to help you get started. The sample dialer application demonstrate how to use the SDK to connect to a SIP service and allows the user to call through the SIP service using a dial pad or a contact list.

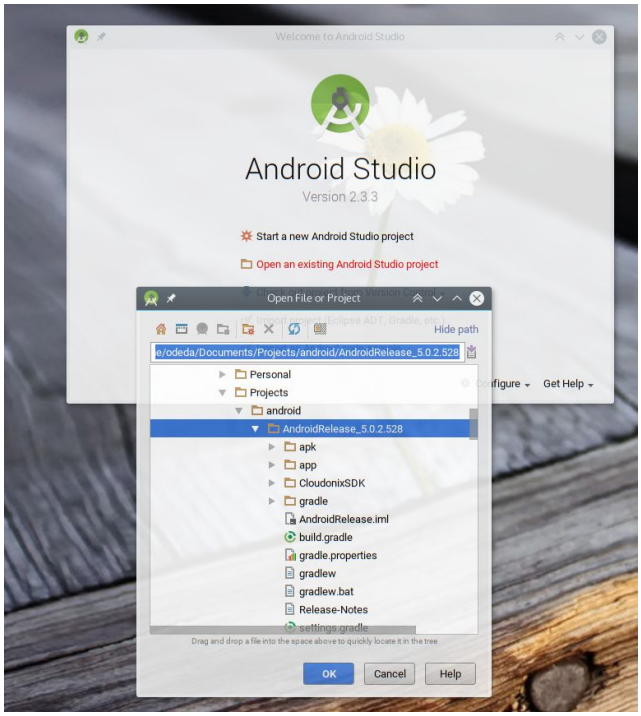
The following sections cover setting up the Cloudonix Mobile SDK sample application and running it.

Android

Please note that Cloudonix Mobile SDK release 5.0 is supported on Android Studio up to version 2.3.3.

Project Setup

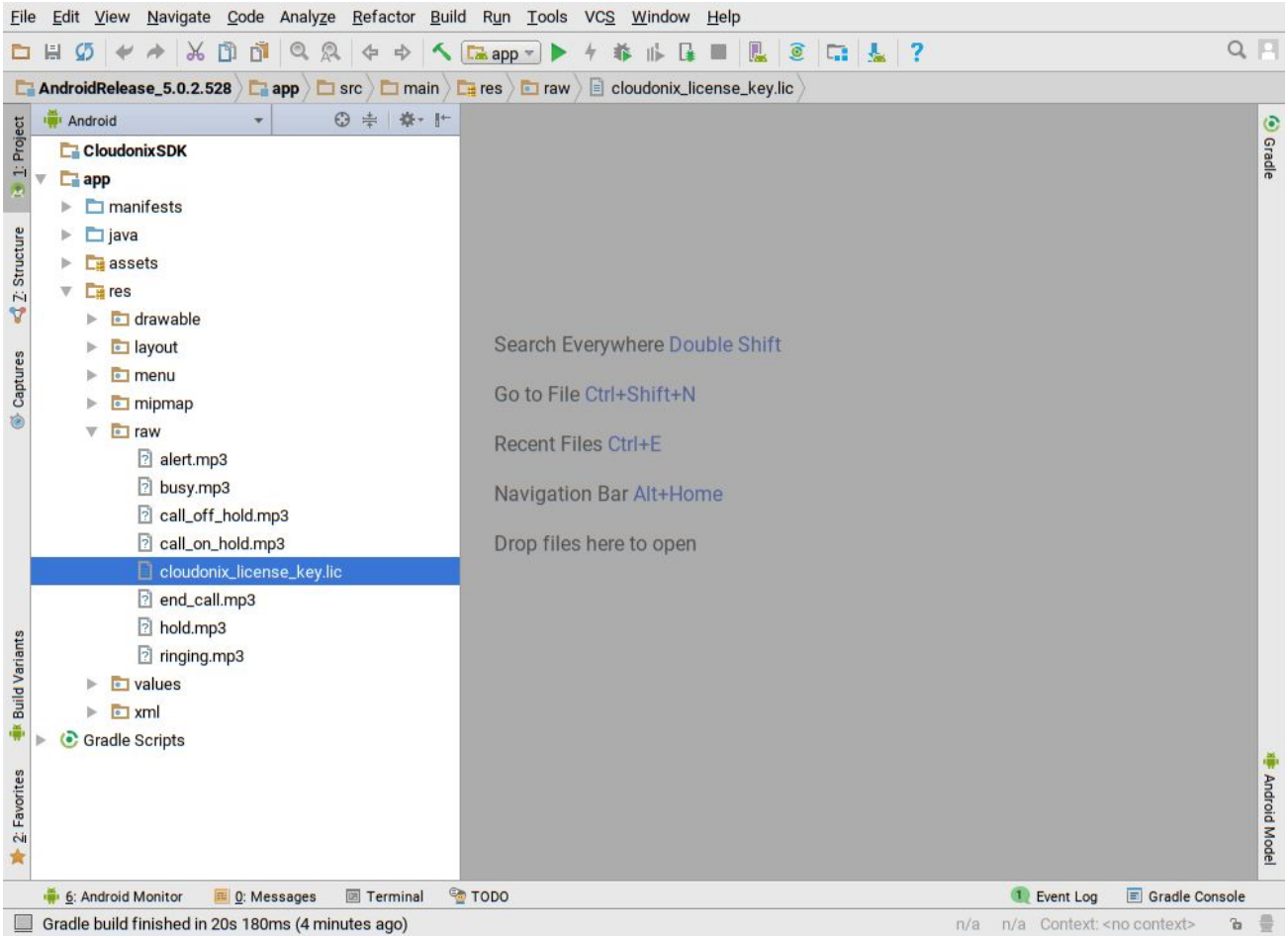
Download the release package and extract it in your projects folder. Then use the Android Studio to open the project:



Tel: +972-73-2557799, Fax:+972-73-2557203, E-mail: info@cloudonix.io, P.O box 199, Udim, Israel 42905

Deploying License Key

After the project loads, navigate to `app` > `res` > `raw` , and replace the empty place-holder file `cloudonix_license_key.lic` with the license file you've received from Cloudonix.

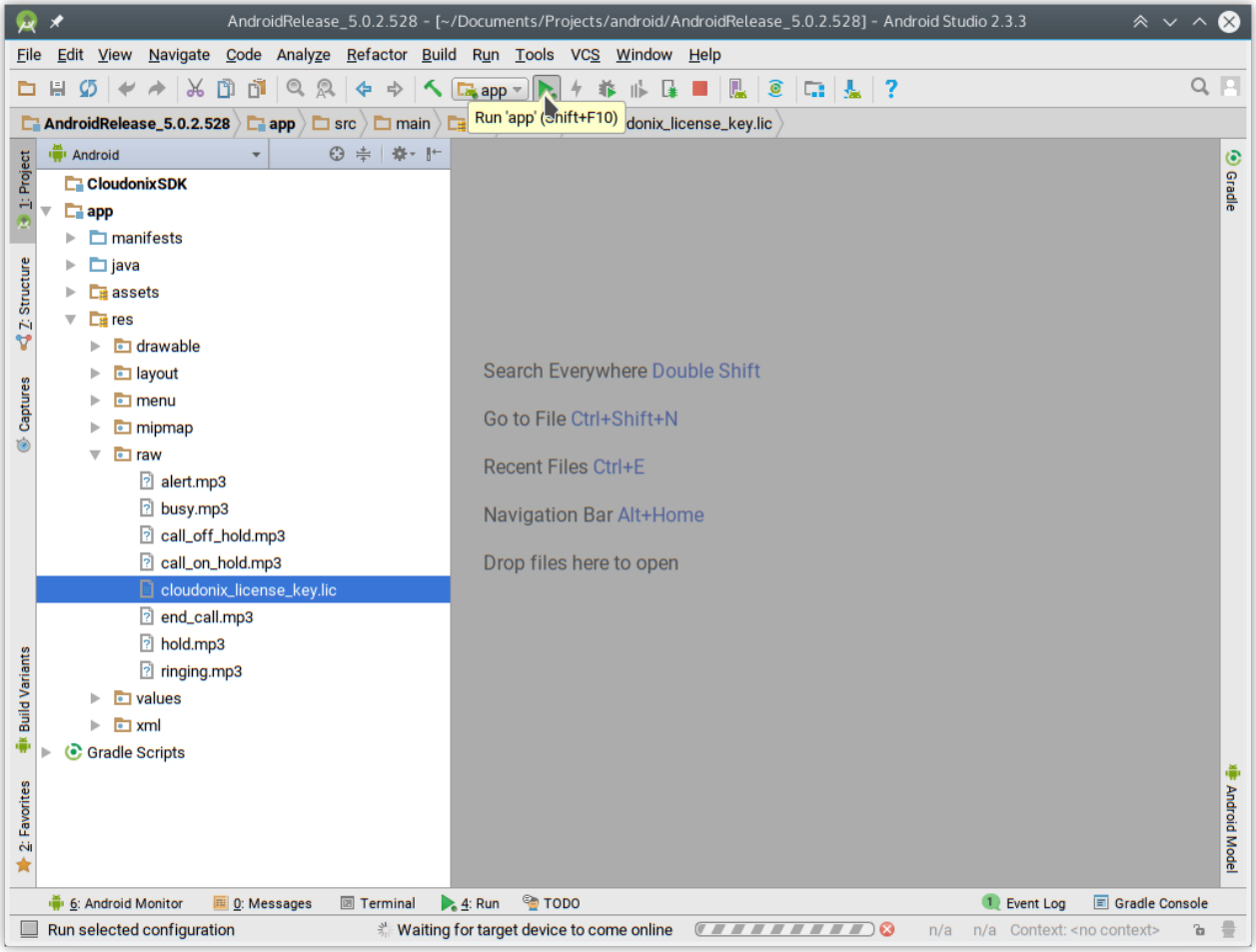


The file name should either be kept the same as in the sample project, or otherwise you can change the name of the file loaded by the sample application by editing the `loadLicenseKey()` method in the `net.Cloudonix.iotech.cloudonixsdk.cloudonixdialer.utils.VoIPClient` class.

Tel: +972-73-2557799, Fax:+972-73-2557203, E-mail: info@cloudonix.io, P.O box 199, Udim, Israel 42905

Build and Run Project

At this point you should be able to click the “Run” button to start the sample dialer application on a connected device or an emulated device.

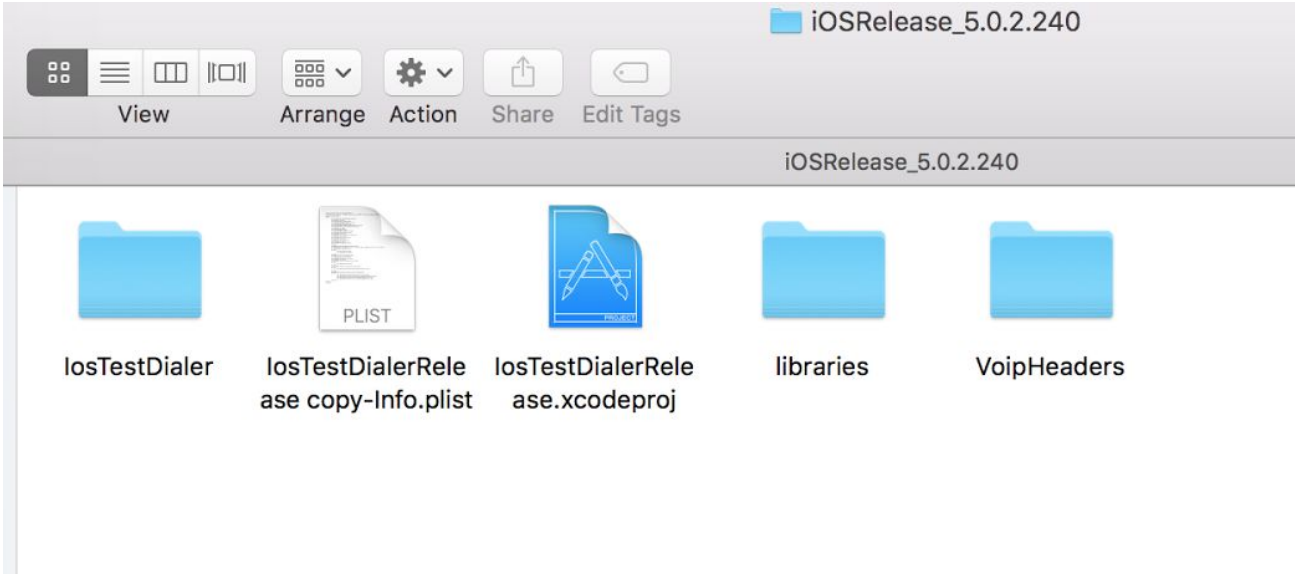


Tel: +972-73-2557799, Fax:+972-73-2557203, E-mail: info@cloudonix.io, P.O box 199, Udim, Israel 42905

iOS

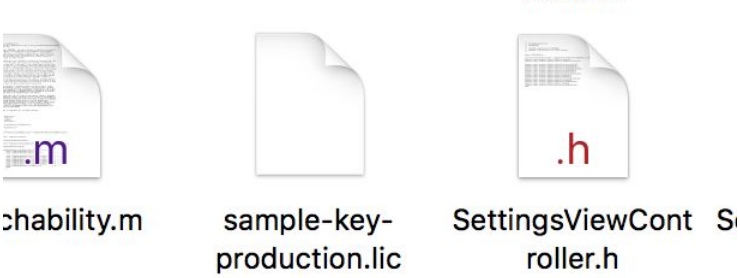
Project Setup

Download the release package and extract it in your projects folder.



Deploying License Key

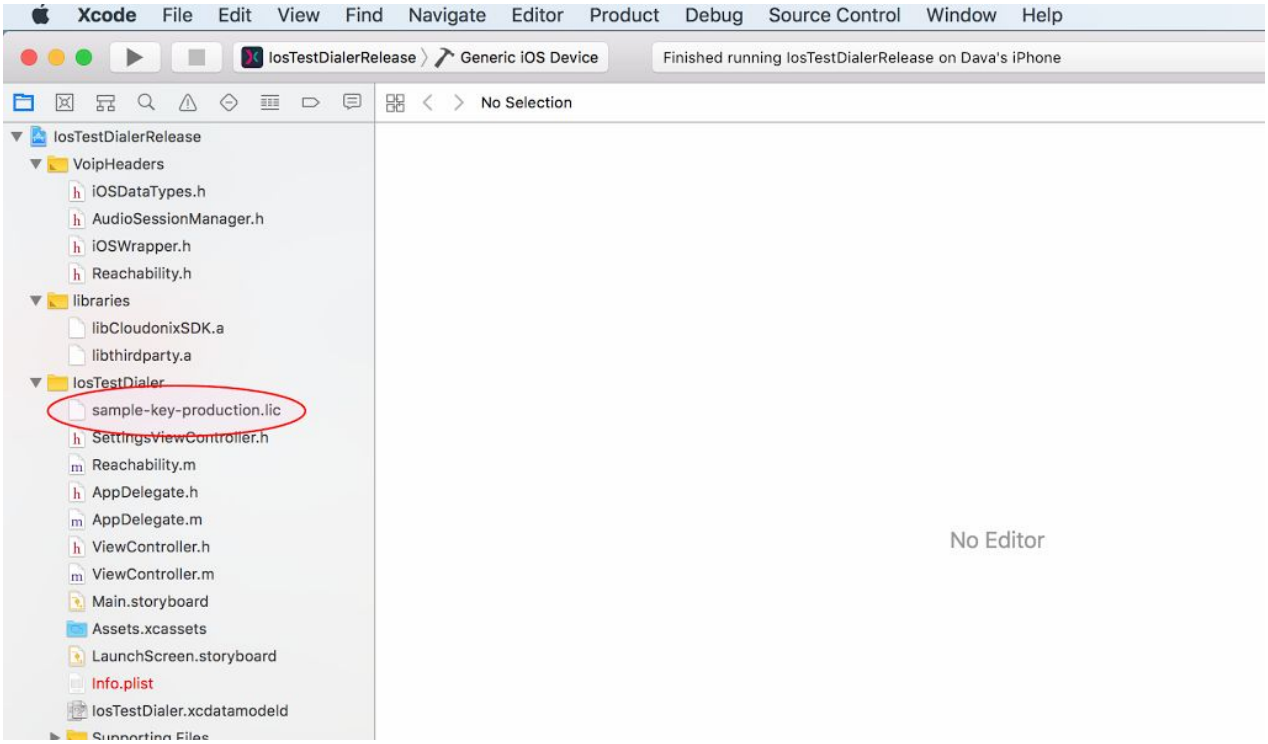
You should have an `IosTestDialer` folder in the project directory, into which the license key file should be copied, and renamed to `sample-key-production.lic`



Build and Run Project

Open the sample application project file `IosTestDialerRelease.xcodeproj` to open the project in Xcode, then use the “Build and Run” command from Xcode:

Tel: +972-73-2557799, Fax:+972-73-2557203, E-mail: info@cloudonix.io, P.O box 199, Udim, Israel 42905



Application Workflows

Initialization

In order to use the Cloudonix Mobile SDK in your application, it must first be initialized with the license key (which you receive from Cloudonix, either a trial or a production license key). The application would load the license key into the Cloudonix Mobile SDK in order to retrieve an instance to the Cloudonix Mobile SDK control object (commonly called “instance” in this guide). This process will verify the license, configure the SDK for the device by utilizing Cloudonix online device configuration service and will eventually issue the `onLicense` event, which can be monitored by the application to check if the licensing process completed successfully. If the licensing process failed - for example if the license key is corrupt or has expired - then the SDK will not be operational.

After initializing the SDK, it needs to be configured with configuration parameters such as allowed codecs and transports, in addition to setting the SIP account details that will be used for calling and receiving calls.

Android

After receiving the SDK client instance, the application should perform these additional steps:

1. Call `addEventsListener()` to be able to receive events from the SDK using callbacks on a `IVoIPObserver` instance.
2. Call `initPreferences()` which is needed for the Android Cloudonix Mobile SDK service to be able to handle incoming calls to the application even when it isn't in the foreground.
3. Use `setConfig()` to set up the SDK.
4. Call `checkBinder()` to see if the SDK service is still running and can be reused. If the service is running, this call will attach to the existing service.
5. If `checkBinder()` returned `false`, the application needs to call `bind()` to create and connect a new instance of the SDK service. After that process is complete, the SDK will send the `onSipStarted` event to let the application know that the service is ready.
6. Once `onSipStarted` is called, the application should set up the SIP account details.

Example Initialization

```
public class VoIPClient implements IVoIPObserver {
  CloudonixSDKClient instance;
…
  Private initSDK(Context ctx) {
    InputStream input = ctx.getResources().openRawResource(R.raw.license);
    try {
      String lic = new String(CryptUtils.convertStreamToByteArray(input));
      instance = CloudonixSDKClient.getInstance(lic);
      instance.addEventsListener(this);
      instance.initPreferences(ctx);
      setConfiguration();
      if (instance.checkBinder())
        return;
      instance.bind(); // will cause onSipStarted to be called
    } catch (IOException e) {
      e.printStackTrace();
    }
  }

  public void setConfiguration() {
    instance.setConfig(ConfigurationKey.USER_AGENT, "MyApp/1.0");
    …
  }

  @Override
  public void onLicense(LicensingState state, String description) {
    // handle licensing problems
  }

  @Override
  public void onSipStarted() {
    instance.setAccount(new RegistrationData() {{
      setServerUrl("sip.server.com");
      …
    }});
  }
```


Tel: +972-73-2557799, Fax:+972-73-2557203, E-mail: info@cloudonix.io, P.O box 199, Udim, Israel 42905

```
}
```

Registration and Registration-Free Setup

The Cloudonix Mobile SDK supports both classic SIP accounts using periodic REGISTER messages to maintain “connection” to the server, as well as Cloudonix Registration-Free mode.

To set up a classic SIP registration mode, call `setConfig(ConfigurationKey.USE_REGISTRATION, "1")` during the configuration process. This is the default, so this step is optional.

To set up Cloudonix Registration-Free mode when using the Cloudonix CPaaS as your SIP service, call `setConfig(ConfigurationKey.USE_REGISTRATION, "0")` during the configuration process.

After setting the account details using `setAccount()`, the Cloudonix Mobile SDK will automatically register with the SIP service if needed, and will issue the on`RegisterState` event whenever the registration state changes. The application can call `isRegistered()` to check the registration status.

Android Example

```
public void onSipStarted() {
  instance.setAccount(new RegistrationData() {{
      setServerUrl("sip.server.com");
      …
  }});
}

@Override
public void onRegisterState(RegisterState result, int expiry) {
  switch(result) {
  case REGISTRATION_SUCCESS: logger.d(TAG, "registered"); break;
  case REGISTRATION_ERROR_CREDENTIALS: logger.d(TAG, "auth error"); break;
  case REGISTRATION_UNREGISTERED: logger.d(TAG, "No longer registered"); break;
  }
}
```

Dialing

After the application completes setting up the SDK and configuring the SIP account details, the application may use the `dial()` command to start a SIP session.

The dial command will start the calling process and will issue on`CallState` events for each stage in the call progress.

Android Example

```
public void dial(String number) {
  dial(number);
}

@Override
public void onCallState(String callId, CallState state, String contactUrl) {
  Switch (callState) {
  …
  }
}
```

Call Reception

When a call is received, the onCallState event will be invoked with the CallState set to CALL_STATE_INCOMING. When the application handles the event, it can call either answer(callId) or reject(callId) as required.

Application Shutdown

It is important to note that when the application is moved to the background, the SDK will still maintain context to allow the application to receive calls. If the application needs to shutdown completely and not receive any calls, it should call destroy(), after which the SDK is completely shut down and in order to start it again it needs to be reinitialized.

Reference

SDK Integration Notes

Android

iOS - Xcode

The Cloudonix Mobile SDK uses the following libraries from the iOS SDK:

- VideoToolbox.framework
- GLKit.framework
- libstdc++.tbd
- libicucore.tbd

Tel: +972-73-2557799, Fax:+972-73-2557203, E-mail: info@cloudonix.io, P.O box 199, Udim, Israel 42905

While these libraries are not automatically linked in to the project, they are available through the iOS SDK.

To link the project with these libraries (in case the application isn't already using them):

1. In the project's setting, change to the "Build Phases" view.
2. In the "Link Binary With Libraries" section, click the plus sign to add a new library.
3. Add each of the libraries listed above.

Tel: +972-73-2557799, Fax:+972-73-2557203, E-mail: info@cloudonix.io, P.O box 199, Udim, Israel 42905