**Creating a new Project with Chat SDK pre-integrated**

**Quick start guide - it takes about 10 minutes! This instruction manual assumes that you are a beginner at using Android Studio and assumes that you want to add Chat SDK to a blank Android Studio project. If you are an advanced user or if you want to add the Chat SDK to an existing project, please use the manual here; NEED LINK!**

1. In Android Studio Go to **File** -> **New** -> **Project**.
2. Enter a name for the **Application** as you see fit. Change the **Company domain** to a relevant value for your organization. Be sure to take a note of the **Application name**, the **Company domain**, and the **Package name**. You will need this information later. Click **Next**. Make sure the checkbox of Phone and Tablet is checked, and then click **Next**. Click **Add No Activity** and click **Finish**.
3. Open the top level `build.Gradle` file. You can do this by clicking on the vertical **Project** tab in the upper left hand corner, then clicking on the horizontal **Project** option in the drop down menu beside it. Click on the folder with the **name of your App**, then click on the **build.gradle** file. When you open it, the tab should have the name of your App. That’s how you know it’s the project level build.gradle file. Find the section of repositories in allprojects, and add the following code inside of it:

Show them what it should look like when its done

maven { url "http://dl.bintray.com/chat-sdk/chat-sdk-android" }  
maven { url "https://maven.google.com" }  
maven { url "https://jitpack.io" }

1. Then add this to your dependencies area of the same file:

classpath 'com.google.gms:google-services:4.0.1'

1. Move your mouse over that line lines slowly, if android studio tells you that the version is outdated, enter the number of the latest version in place of the 4.0.1.
2. Now go to your app level build.gradle file. Click on the **app** folder above the **build.gradle** file on the right, and then open the **build.gradle** file in it. The file should have the title "app" when you open it.
3. Add the following code to the build.gradle file, in the section dependencies:

implementation 'co.chatsdk.chatsdk:chat-sdk-firebase-push:4.1.35'  
implementation 'co.chatsdk.chatsdk:chat-sdk-firebase-adapter:4.1. 35'  
implementation 'co.chatsdk.chatsdk:chat-sdk-firebase-file-storage:4.1. 35'  
implementation 'co.chatsdk.chatsdk:chat-sdk-core:4.1. 35'  
implementation 'co.chatsdk.chatsdk:chat-sdk-firebase-push:4.1. 35'  
implementation 'co.chatsdk.chatsdk:chat-sdk-firebase-ui:4.1.26'  
implementation 'com.google.firebase:firebase-auth:16.0.3' check this line  
implementation 'com.google.android.gms:play-services-auth:16.0.0' check this line

1. Move your mouse over these lines slowly, if android studio tells you that these versions are outdated, enter the number of the latest version in the appropriate line in place of the number of the latest version. They can also check the latest project to see the latest Chat SDK version number
2. Find the android { } section of the file Add this code…. It will now look like this ….

compileOptions {  
  sourceCompatibility JavaVersion.VERSION\_1\_8  
  targetCompatibility JavaVersion.VERSION\_1\_8  
}

1. Add this to the very end of the app level build.gradle file:

apply plugin: 'com.google.gms.google-services'

1. Now you need to create a new class. Inside the **app** folder on the left, click on **src**, then on main, and then on **java**. Under **java** there should be a folder with the package name. Right click on it, then go to **new** and click on **Java Class**. Call the class "AndroidApp", or any other name you desire, under the label Superclass, write "Application". Click \*\*Ok\*\*. In the body of the class, erase all text **except for the first line.** This would normally be package PACKAGE NAME;and copy this code into it:

import android.app.Application;  
import android.content.Context;  
​  
import co.chatsdk.core.error.ChatSDKException;  
import co.chatsdk.core.session.ChatSDK;  
import co.chatsdk.core.session.Configuration;  
import co.chatsdk.firebase.FirebaseNetworkAdapter;  
import co.chatsdk.firebase.file\_storage.FirebaseFileStorageModule;  
import co.chatsdk.firebase.push.FirebasePushModule;  
import co.chatsdk.ui.manager.BaseInterfaceAdapter;  
​  
public class AndroidApp extends Application {  
​  
  @Override  
  public void onCreate() {  
      super.onCreate();  
​  
      Context context = getApplicationContext();  
​  
// Create a new configuration  
      Configuration.Builder config = new Configuration.Builder(context);  
​  
// Perform any configuration steps (optional)  
      builder.firebaseRootPath("prod");  
​  
// Initialize the Chat SDK  
      try {  
          ChatSDK.initialize(builder.build(), new BaseInterfaceAdapter(context), new FirebaseNetworkAdapter());  
      }  
      catch (ChatSDKException e) {  
      }  
         
builder.googleMaps("");   
​  
// File storage is needed for profile image upload and image messages  
      FirebaseFileStorageModule.activate();  
      FirebasePushModule.activateForFirebase();  
​  
// Activate any other modules you need.  
// ...  
​  
  }  
}  
​

1. Open your AndroidManifest.xml file, it should be in the "main" folder. Add this code to the <application section: android:name=".AndroidApp". Currently, your Android Manifest.xml file should look something like this:

<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
  package="PACKAGE NAME">  
​  
  <application  
      android:name=".AndroidApp"  
      android:allowBackup="false"  
      android:icon="@mipmap/ic\_launcher"  
      android:label="APP NAME"  
      android:roundIcon="@mipmap/ic\_launcher\_round"  
      android:supportsRtl="true"  
      android:theme="@style/AppTheme"/>  
</manifest>

1. On the line android:theme="@style/AppTheme"/> delete the / then click after the > and hit the enter button. Now write </application>. Copy the code below, then click to the right of the > in the line android:theme="@style/AppTheme">, hit enter and paste the code :

<activity android:name="co.chatsdk.ui.login.LoginActivity">  
  <intent-filter>  
      <action android:name="android.intent.action.MAIN" />  
      <category android:name="android.intent.category.LAUNCHER" />  
  </intent-filter>  
</activity>

1. Now add the line <?xml version="1.0" encoding="utf-8"?> At the very top of the file. The result should look like this:

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
  package="PACKAGE NAME">  
​  
  <application  
      android:name=".AndroidApp"  
      android:allowBackup="true"  
      android:icon="@mipmap/ic\_launcher"  
      android:label="APP NAME"  
      android:roundIcon="@mipmap/ic\_launcher\_round"  
      android:supportsRtl="true"  
      android:theme="@style/AppTheme">  
      <activity android:name="co.chatsdk.ui.login.LoginActivity">  
          <intent-filter>  
              <action android:name="android.intent.action.MAIN" />  
              <category android:name="android.intent.category.LAUNCHER" />  
          </intent-filter>  
      </activity>  
  </application>  
</manifest>

1. The purpose of this step was to set the Chat SDK login activity to launch when the app is launched, meaning that this login screen will be the first thing you see when you run the App.
2. Open your Android Studio Suite. Go to the very top right hand button (When you mouse over this button, it will say sign in to Google, and use it to sign in to Google with your Google account. If you do not have a Google account, you can use the button to create one.
3. Now click on the button called **Sync Project with Gradle Files**. It should be at the top right hand corner, near the google account button. Ignore any messages telling you that the build failed,
4. Go to **Tools** -> **Firebase**. Go to the tab on the right, click on analytics, click on **Log an Analytics event**, and then click **Connect to Firebase** then click on **Connect to firebase**. If there are errors, click **Connect to firebase** again and click **sync**, until the button turns into the word "Connected".
5. Now you can go to the [Firebase Console](https://console.firebase.google.com/) in your web browser, and you should find your project. It should be a large white tile with the name of your app. Click on your project, then go to the firebase dashboard, and go to **Authentication** -> **Sign-In-Method**, and click on whichever sign in options you like. We recommend clicking only on the **Sign in with Email and Password** option. Switch both Sign in switches to "On" and click **Save**.
6. Get the push token. Click on the **Gear Button** in the top left corner, and then the **Project Settings** button. In the **General** tab, Cloud messaging server key.
7. Now go back to Android Studio. Add the following to the setup code in the AndroidApp's onCreate method.

builder.firebaseCloudMessagingServerKey("YOUR CLOUD Server Key");

1. Go back to your [Firebase Console](https://console.firebase.google.com/) , click on your app, Click on **Storage** at the left, click on **Get Started**, then click on **Got it**.
2. Go back to your [Firebase Console](https://console.firebase.google.com/) , click on your app, Click on **Database, create realtime database, start in locked mode. Click the rules tab.** Delete everything in the box, then go to this [rules.json](https://github.com/chat-sdk/chat-sdk-ios/blob/master/rules.json) file, copy everything in the box (approximately 355 lines), and paste it into the box in the firebase console. Click on **Publish**.
3. Now click on the button called **Sync Project with Gradle Files**. It should be at the right left hand corner near the google account button. When the gradle sync completes, your App is ready to go!

Google maps

Google places (Following link on main docs), get started, check places check box, Continue, Select project and then Agree to terms and conditions, Click next, Set up a billing account. Say to do this you need a Google Billing account. If you want to do that, then continue, otherwise disable location messages… using… Although you need to setup billing, Google give you 200 USD per month for free. So you can load 10 million free location messages for free per month

We need to see where to get the final key

I think its worth you to add your payment details**Conclusion**

Congratulations! 🎉🎉 You've just turned your app into a fully featured instant messenger! Keep reading below to learn how to further customize the Chat SDK as well as add various other modules as needed.