**HƯỚNG DẪN SỬ DỤNG VÀ TÍCH HỢP**

**DỊCH VỤ FIO (CALL/CHAT) TRONG ỨNG DỤNG ANDROID MOBILE**

*Ghi chú: Một số từ ngữ chuyên ngành sẽ không được dịch ra Tiếng Việt để giữ đúng ý nghĩa của từ.*

1. **GIỚI THIỆU**

FIO là nền tảng OTT/Message cung cấp thư viện (SDK) & API để giúp cho nhà phát triển dễ dàng tích hợp tính năng gọi điện (call) / trò chuyện (chat) vào trong các ứng dụng mobile.

FIO SDK được phát triển bởi MIDEAS Co., ltd với đội ngũ nhân viên nhiều có năm kinh nghiệm trong lĩnh vực Mobile và công nghệ.

Dưới đây là hướng dẫn chi tiết (step by step) giúp cho việc tích hợp FIO SDK vào dự án android mobile app của bạn.

Ghi chú: Để có thể chạy thử nghiệm các dự án mẫu, điều đầu tiên bạn cần phải cài đặt Android Studio (link cài đặt bên dưới):

1. Clone sample project from git hub:

https://github.com/mideasvn/FIO-SDK

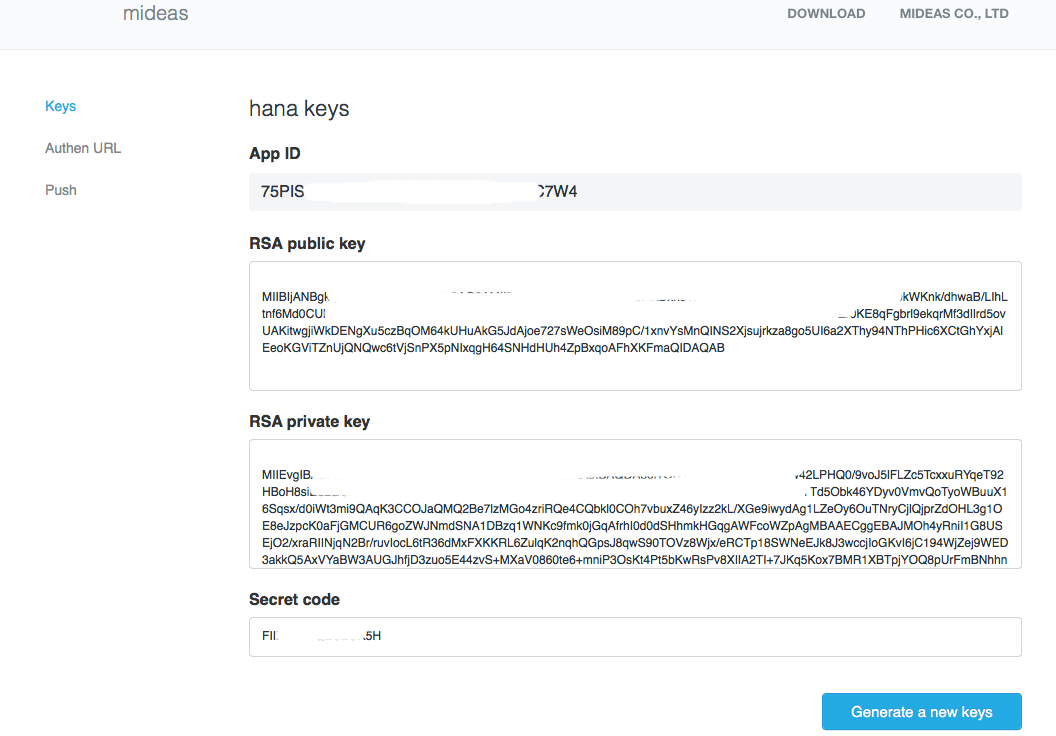
1. Open and run in Android Studio
2. **CÀI ĐẶT**

Để cài đặt SDK lên ứng dụng của bạn, bạn cần :

B1: Khởi tạo 1 tài khoản FIO (https://fio.mideasvn.com/developers/signin).

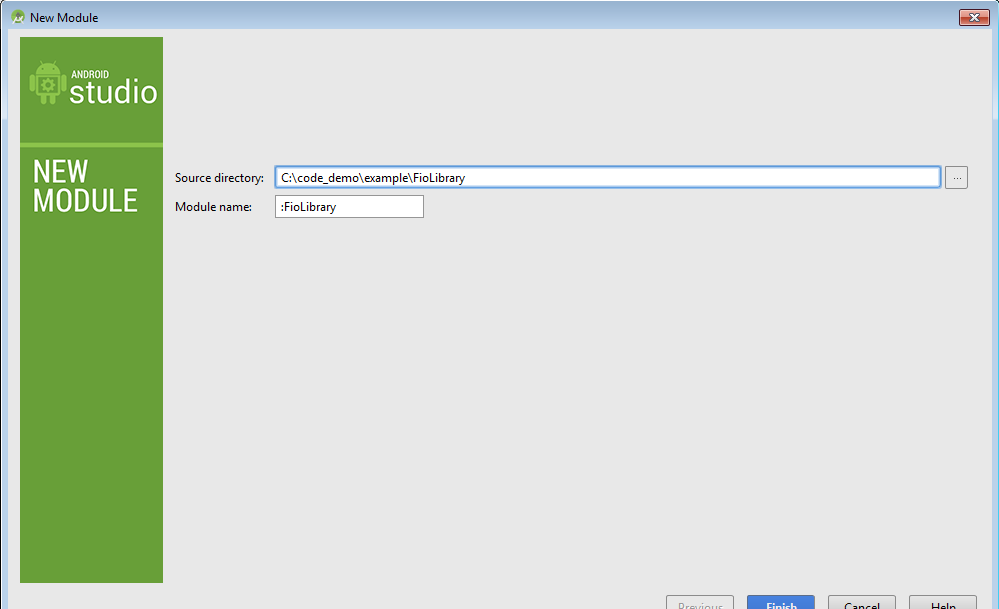
B2: Sau khi tạo tài khoản thành công , bạn cần tạo một project . Khi project được tạo ra, Mideas cung cấp cho bạn :

* Một ID ứng dụng ( AppID )
* Một cặp khóa RSA (khóa công khai được sử dụng trên ứng dụng của bạn, chìa khóa riêng được sử dụng trên máy chủ của bạn ) .



Các hướng dẫn sau đây sẽ giúp bạn tích hợp FIO SDK vào ứng dụng của bạn:

* 1. Module library:
  + Inport module library Mideas vào project



* + Cài đặt trong file build.gradle:

defaultConfig {

minSdkVersion 16

multiDexEnabled true

}

dexOptions {

javaMaxHeapSize "4g”

}

}

repositories {

flatDir {

dirs 'libs', "../FioLibrary/libs"

}

}

dependencies {

compile fileTree(dir: 'libs', include: ['\*.jar'])

compile 'com.android.support:support-v4:23.2.1'

compile 'com.android.support:appcompat-v7:23.4.0'

compile project(':FioLibrary')

compile 'com.android.support:multidex:1.0.0'

}

* 1. AndroidManifest.xml
  + FIO SDK đòi hỏi một số permissions và references (gcm , google map api key ...) được cài đặt trong tập tin AndroidManifest.xml ở ứng dụng của bạn. Những điều khoản đó cho phép SDK để giám sát trạng thái mạng và nhận tin nhắn từ Google Cloud Messaging ... Dưới đây là một ví dụ với một gói com.example. Lớp Application.java của bạn phải extends FIOApplication.

<**application  
 android:name=".CusApplication"  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 *<!-- META -DATA -->* <**meta-data  
 android:name="com.google.android.gms.version"  
 android:value="@integer/google\_play\_services\_version"** />  
 <**meta-data  
 android:name="com.google.android.maps.v2.API\_KEY"  
 android:value="---------------------------------"** />  
  
 <**activity android:name=".MainActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
</**application**>

* + Cài đặt Google Maps API key: FIO SDK có sử dụng Google Maps API cho tính năng chia sẽ vị trí trong chat. Cần thiết lập chứng thực từ Google Maps API tham khảo các liên kết sau đây để lấy được keys api của riêng bạn và thay thế nó trong AndroidManifest.xml:

<https://developers.google.com/maps/documentation/android-api/signup>

1. **KẾT NỐI & TÍCH HỢP:**
   1. Khởi tạo dịch vụ với appId và publicKey đã đăng ký

Sử dụng appId và publicKey đã được cấp ở “Phần 2” khi đăng ký ứng dụng trên website https://fio.mideasvn.com/developers/signin để khởi tạo dịch vụ. Ví dụ:

String appId = " S6T9XF9XY4YU8TG8ZZDLETQQ8FKCQVCX ";

String publicKey = "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAk4ImxI2wHc0PtifGvZimBPhnJrGo046eAFoCejPdFO95Kgcg7oZaxZItfAPHASPK17pn4AHU8WjPphVLJUFMbmBS3oPUpRGip63hvLZvkbJxJRMcf2DtPItZnftqGjAZ23g1ANw11g5l1Xo0eDlhSe6Xgcfw46fhY4/uKVY+1N343HqB4knNhhKS/QdEh6JUneE2wroA7UyUov4n+CHPCaDwR2VpXrQdHNtiQ8sHMzBVXUygryszDc/F2TS213HknCCy6Fx1TzBqTZDzvpak9OJvVWYIqFQncxlk0yhVRrajBtITC4I0lI3+eMEq1mh5KHOCmCtBTFYgMocVEkNXowIDAQAB "

* + FIO SDK có một FioClient giao diện chính để tương tác với dịch vụ khác trong FIO. Bạn phải khởi tạo đối tượng FioClient (chỉ 1 lần duy nhất) trong hàm khởi tạo chính onCreate().

FIOClinet.newInstance(cusApplication, Constant.appId, Constant.publicKey);

* 1. Listeners
  + Để có thể lắng nghe và nhận các tin nhắn cũng như cuộc gọi từ người dùng khác, bạn phải đăng ký 2 phương thức lắng nghe FioConnectListener, FioUserListener khi khởi tạo đối tượng FioClient.

FIOClinet.getInstance().setXmppConnectivityChangeListener(**new** XmppConnectivityChangeListener());

FIOClinet.getInstance().setFioUserListener(**new** FioUserListener());

* 1. Kết nối và chứng thực
  + In order to a user to chat or call, you must authenticate them first. Mideas will accept any unique String as a User ID (UIDs, email addresses, phone numbers, usernames, etc), which we call as FioUserId. So you can use any new or existing User Management system. The process is described in the following model:
* Khởi đầu: Đăng ký tài khoản và nhận key chứng thực tại https://fio.mideasvn.com/developers/signin

a. Tạo ứng dụng (project)

b. Tạo keys chứng thực

c. Cập nhật URL để xác thực thông tin user giữa hệ thống của bạn với hệ thống MIDEAS.

* Bước 1: Khởi tạo appID và public key (như mô tả tại mục 3.1)
* Bước 2: Khởi động kết nối với việc sử dụng phương thức fioClient.connect(String userId, String userCredentials)

fioClient.connect(String userId, String userCredentials,String name);

Parameter: userId: your user id on your system, as described above

userCredentials: can be your password, token on your system. Mideas will not use this information. It's forwared to your server to authenticate your user.

name: optional, used to display on push notification.

Bạn phải đăng ký lắng nghe sự kiện kết nối và chức thực FioConnectListener, các phương thức chủ yếu: connected, disconnected, connect failed...

FIOClinet.getInstance().setXmppConnectivityChangeListener(**new** XmppConnectivityChangeListener() {  
 @Override  
 **public void** onXMPPConnected() { *// Connect and authenticate successfully, can start chatting, calling from here* runOnUiThread(**new** Runnable() {  
 @Override  
 **public void** run() {}  
 });  
 }  
  
 @Override  
 **public void** onXMPPDisconnected() { *// Disconnected from Mideas service* runOnUiThread(**new** Runnable() {  
 @Override  
 **public void** run() {}  
 });  
 }  
 @Override  
 **public void** onXMPPConnecting() {  
 runOnUiThread(**new** Runnable() {  
 @Override  
 **public void** run() {}  
 });  
 }  
 @Override  
 **public void** onXmppPrepareConnecting(**long** intervalTimeToConnecting) {}  
});

* Step 3:
* Step 4:
* Step 5:
  1. Lớp FioUserListener

FioUserListener phải được đăng ký ngay khi khởi tạo đối tượng FioClient. Nó sử dụng để lấy các thông tin cơ bản của người sử dụng ứng dụng, bao gồm: name, avatar... để hiển thị trên màn hình chat hoặc gọi điện.

FIOClinet.getInstance().setFioUserListener(**new** FioUserListener() {  
 @Override  
 **public void** getUserInfo(List<String> userIds) {  
  
 }  
  
 @Override  
 **public** JSONArray getListFriend() {  
 JSONArray arrayContact = **new** JSONArray();  
 **try** {  
 **for** (**int** i = 0; i < Constant.listFrienUserId.length; i++) {  
 JSONObject obContact = **new** JSONObject();  
 obContact.put(**"cmsisdn"**, Constant.listFrienUserId[i]);  
 obContact.put(**"cname"**, Constant.listFriendName[i]);  
 arrayContact.put(obContact);  
 }  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 **return** arrayContact;  
 }  
}

Trong hàm JSONArray getListFriend() có thể return null để lấy danh sách bạn bè từ server Mideas hoặc truyền vào danh sách bạn bè của bạn như trong ví dụ.

1. **API HỖ TRỢ NHÀ PHÁT TRIỂN**
   1. Chat với 1 user

fioClient.startChat(Context context, String userId);

Parameter: context: The context to start chat, required

userId: identifier of app’s user, required

* 1. Call tới 1 user

fioClient.startCall(Context context, String userId);

Parameter: context: The context to start call, required

userId: identifier of app’s user, required

* 1. Tạo 1 cuộc chat nhóm

fioClient.createGroup(Context context);

Parameter: context: The context to create group, required

* 1. Lịch sử đã chat

fioClient. viewChatHistory(Context context);

Parameter: context: The context to view chat history, required

* 1. Lịch sử cuộc gọi

fioClient. viewCallHistory(Context context);

Parameter: context: The context to view call history, required

* 1. Danh bạ người dùng

fioClient. viewCaontact(Context context);

Parameter: context: The context to view contact, required

* 1. Cài đặt

fioClient. viewSetting(Context context);

Parameter: context: The context to view setting, required

* 1. Thông tin 1 user

fioClient.viewUserInfo(Context context);

Parameter: context: The context to view info user, required

* 1. Dừng kết nối tới server

Dừng kết nối tới server FIO service và thoát khỏi hệ thống

fioClient. disconnect();