



Kakao 2.0 SDK Guide with VOLTUS SDK

Version History

[illegible]

1. Introduction

해당 문서는 VOLTUS SDK와 Kakao 2.0 SDK와의 적용을 위한 가이드를 제시하는 문서이다. VOLTUS SDK 에서 제공되는 Kakao 관련 API를 주로 소개하며, Kakao와 관련한 세부적인 사항은 아래의 주소에서 열람이 가능하다.

Kakao 2.0 Development Guide : <https://gamecenter.kakao.com/docs/documents/243>

1.1 개발 환경 설정

Android 및 iOS 플랫폼에서의 각 개발 관련 설정을 가이드한다.
세부적인 Kakao 관련 개발 설정은 아래의 주소에서 확인 가능하다.

<https://gamecenter.kakao.com/docs/documents/281>

1.2 Android 설정

1.2.1 각 lib 프로젝트 링크

제공되는 SDK 폴더에서 curo-sdk_kakao, kakao_android_sdk_for_eclipse, google-play-services_lib 프로젝트를 이클립스 등의 IDE에서 임포트한다. 기존에 facebook-sdk lib 프로젝트가 있는 경우에는 kakao 용 VOLTUS SDK에서는 사용하지 않으므로 해당 프로젝트는 필요하지 않다.

1.2.2 게임센터 프로젝트 내에 키해시 등록

Admin 페이지 : <https://gamecenter.kakao.com/admin/apps>)

Android 프로젝트의 경우에는 게임센터 Admin 페이지에 디버그 및 릴리즈 키해시 값이 등록되어야 한다.

1.2.3 Eclipse 프로젝트에서 앱키 xml 파일 설정

게임 프로젝트의 res/value 폴더내에 kakao_string.xml 파일을 만든 후 아래의 값을 추가한다.

[kakao_strings.xml]

- The native app key is issued when the app is created. The native app key can be seen at [GameCenter > Admin > Detail Info].
- Define the native key app as "kakao_app_key" and register the app key in 'AndroidManifest.xml'.

```
<resources>
    <string name="kakao_app_key">AAAAAAAAAAAAAAAAAAAAAA</string>
    <string name="kakao_app_key">AAAAAAAAAAAAAAAAAAAAAA</string>
    <string name="kakao_scheme">kakaoAAAAAAAAAAAAAAAAAAAAAA</string>
    <string name="kakaolink_host">kakaolink</string>
    <string name="kakaostory_host">kakaostory</string>
</resources>
```

1.2.4 AndroidManifest.xml 파일 설정

```
<!-- 1. Setup the network auth to communicate with the server. -->
<uses-permission android:name="android.permission.INTERNET" />
```

```

<!-- Auth to access the SD card is needed to cache images. -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

<application
    ....
    android:name="com.kakao.game.sample.common.GlobalApplication">

    <activity
        ...
        android:name=".SampleLoginActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    ...
    <!-- 3. Register the app key as 'com.kakao.sdk.AppKey'. -->
    <meta-data
        android:name="com.kakao.sdk.AppKey"
        android:value="@string/kakao_app_key" />

    <!-- kakaoTalk를 위한 activity -->

    <activity android:name=".kakaotalk.KakaoTalkMainActivity"
        android:launchMode="singleTop"
        android:windowSoftInputMode="adjustResize">
        <intent-filter>
            <action android:name="android.intent.action.VIEW"/>
            <category android:name="android.intent.category.DEFAULT"/>
            <category android:name="android.intent.category.BROWSABLE"/>
            <data android:scheme="@string/kakao_scheme" android:host="@string/kakaolink_host"
        />
        </intent-filter>
    </activity>

    <activity android:name=".kakaotalk.KakaoTalkFriendListActivity"
        android:launchMode="singleTop"
        android:windowSoftInputMode="adjustResize">
    </activity>

    <activity android:name=".kakaotalk.KakaoTalkChatListActivity"
        android:launchMode="singleTop"
        android:windowSoftInputMode="adjustResize">
    </activity>

    <activity
        android:name="com.kakao.game.sample.friends.FriendsMainActivity"
        android:configChanges="keyboardHidden|orientation"
        android:windowSoftInputMode="adjustResize|stateHidden"
        android:launchMode="singleTop"/>

    <activity
        android:name="com.kakao.game.sample.guild.GuildMainActivity"

```

```
android:configChanges="keyboardHidden|orientation"
android:windowSoftInputMode="adjustResize|stateHidden"
android:launchMode="singleTop"/>
```

```
<activity android:name=".kakaostory.StoryMainActivity"
    android:configChanges="keyboardHidden|orientation"
    android:windowSoftInputMode="adjustResize|stateHidden"
    android:launchMode="singleTop"/>
```

```
<activity android:name=".invitaionevent.InvitationEventListActivity"
    android:launchMode="singleTask"/>
```

```
<activity
    android:name="com.kakao.auth.authorization.authcode.KakaoWebViewActivity"
    android:label="@string/com_kakao_login_button"
    android:launchMode="singleTop"
    android:windowSoftInputMode="adjustResize">
```

```
<intent-filter>
    <action android:name="android.intent.action.MAIN"/>
    <category android:name="android.intent.category.DEFAULT"/>
</intent-filter>
```

```
</activity>
```

```
<!-- kakaoTalk를 위한 activity -->
```

```
</application>
```

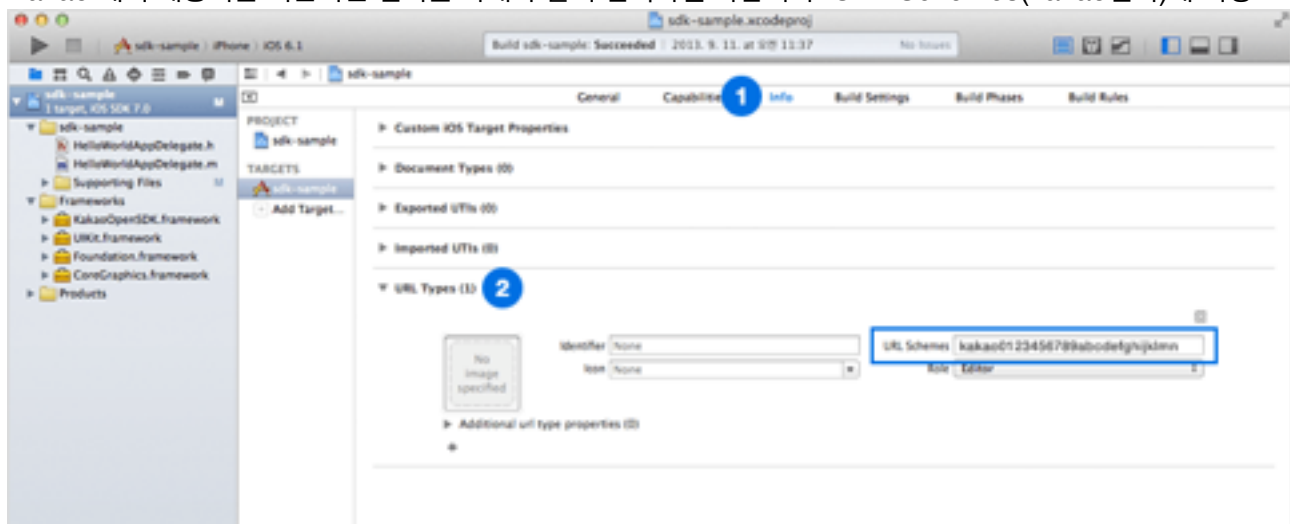
1.3 iOS 설정

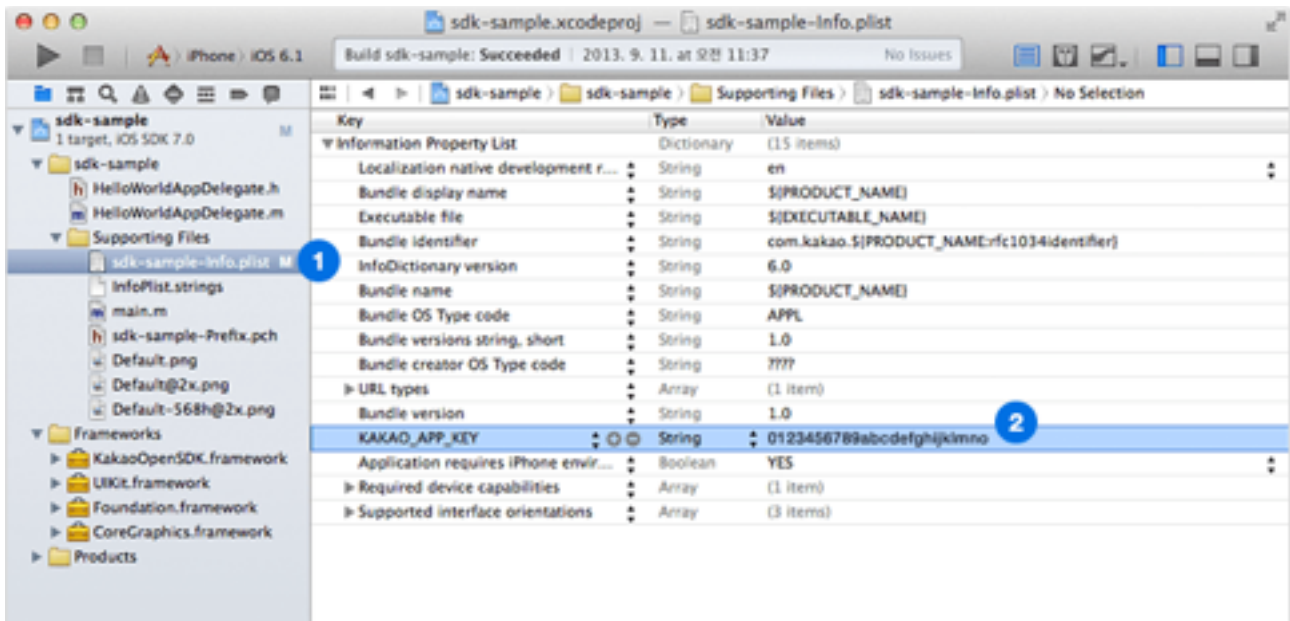
1.3.1 Kakao 관련 framework 파일 импорт

제공되는 SDK 폴더 내에서 KakaoGameSDK.framework, KakaoOpenSDK.framework 파일을 임포트한다. 또한 기본적으로 Kakao SDK 에서 AdSupport.framework 파일도 사용하므로 기존 프로젝트에 임포트되어 있지 않다면 임포트하길 바란다.

1.3.2 앱키 설정

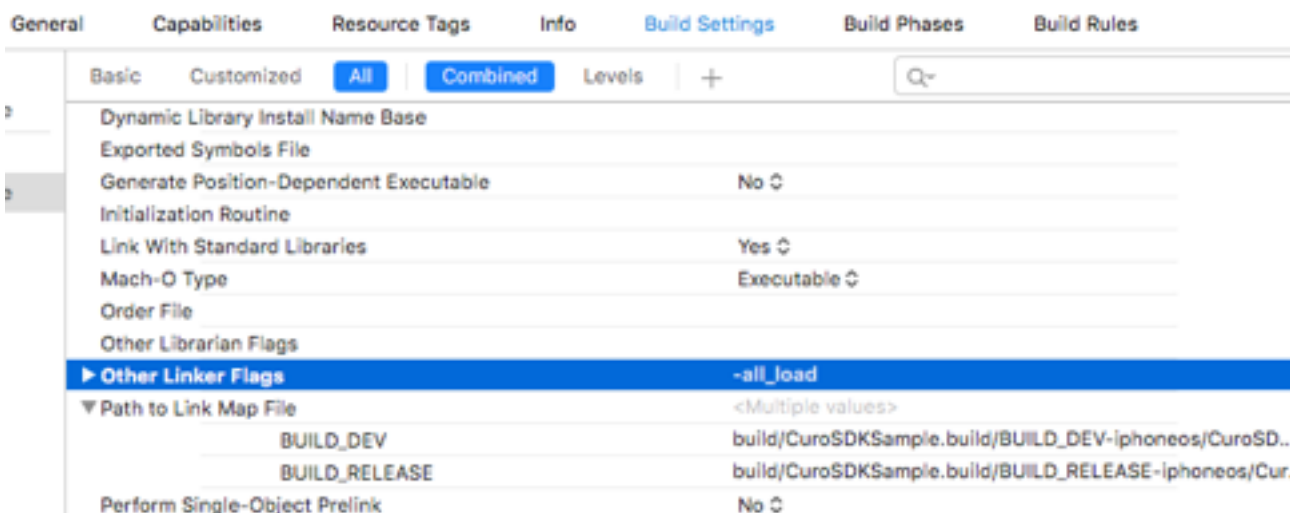
Kakao 에서 제공되는 기본적인 앱키를 아래와 같이 입력하길 바랍니다. URL Schemes(kakao앱키)에 지정





1.3.3 Xcode BuildSetting 설정

Other Linker Flags에 -all_load 추가



1.3.4 iOS9을 위한 도메인 화이트리스트 설정

iOS9 이상에서는 특정 도메인만 네트워크 통신이 가능하므로 아래의 값들처럼 설정하여 특정 주소를 열어주어야 한다.

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSExceptionDomains</key>
  <dict>
    <key>kakao.com</key>
    <dict>
      <key>NSIncludesSubdomains</key>
      <true/>
      <key>NSExceptionRequiresForwardSecrecy</key>
      <false/>
      <key>NSExceptionAllowsInsecureHTTPLoads</key>
```

```

        <true/>
    </dict>
    <key>kakao.co.kr</key>
    <dict>
        <key>NSIncludesSubdomains</key>
        <true/>
        <key>NSEnvironmentRequiresForwardSecrecy</key>
        <false/>
        <key>NSExceptionAllowsInsecureHTTPLoads</key>
        <true/>
    </dict>
    <key>kakaocdn.net</key>
    <dict>
        <key>NSIncludesSubdomains</key>
        <true/>
        <key>NSEnvironmentRequiresForwardSecrecy</key>
        <false/>
        <key>NSExceptionAllowsInsecureHTTPLoads</key>
        <true/>
    </dict>
</dict>
</dict>
</dict>

```

1.3.5 LSAApplicationQueriesSchemes 설정

프로젝트 Info.plist에서 LSAApplicationQueriesSchemes 키 값에 아래와 같은 값을 추가하여야 한다.

```

<key>LSApplicationQueriesSchemes</key>
<array>
    <string>kakao{app_key}</string>
    <string>kakaokompassauth</string>
    <string>storykompassauth</string>
    <string>kakaolink</string>
    <string>kakaotalk-4.5.0</string>
    <string>kakaostory-2.9.0</string>
</array>

```

2. 주요 Kakao 관련 API

Android 및 iOS를 위한 주요 API를 가이드한다.
 각 주요 API에서는 대부분 리턴 코드값을 가지고 있다.
 성공 이외에 코드는 Kakao 게임 센터의 내용을 확인바란다.

Kakao Common Response Code : <https://gamecenter.kakao.com/docs/documents/285>

2.1 Android API

VOLTUS SDK에서 제공되는 주요 Android API를 설명한다.
 좀 더 자세한 Kakao 관련 Android API의 내용은 아래의 주소에서 확인바란다.

<https://gamecenter.kakao.com/docs/documents/287>

2.1.1 Login API

Kakao 로그인을 위해서는 아래와 함수를 호출한다.

기존의 VOLTUS SDK 의 onFinishLogin, onFailedLogin 리스너 함수에서 로그인 여부를 확인할 수 있다.

```
CRSDKManager.loginForKakao();
```

2.1.2 Logout API

Kakao 로그아웃을 위한 API이다.

기존의 VOLTUS SDK 의 onFinishLogout, onFailedLogout 리스너 함수에서 로그인 여부를 확인할 수 있다.

```
CRSDKManager.logoutForKakao();
```

2.1.3 로그인 확인(Verify) API

로그인을 한번 하면 일정 시간 동안에는 AccessToken이 유효하며 자동적으로 로그인을 할 수 있도록 된다.

따라서 게임 내에 로그인 관련 화면에서 로그인 버튼을 표시하기 이전에 현재 로그인 상태가 유효한지 체크하여 유효하면 자동적으로 로그인을 처리를 완료하여 로그인 이후 상태로 진입하면 된다.

아래의 함수는 현재의 로그인 상태를 알려주는 함수이다.

```
boolean isSessionOpen = CRSDKManager.verifyLoginForKakao();

if(isSessionOpen)
{
    CRSDKManager.pushLog("Login Session is already opened.");
}
```

2.1.4 카카오 프로필 API

카카오 관련 프로필을 가져올 수 있는 API 이다.

아래와 같이 호출하면 CRUser 내에 카카오 프로필(uid, name, pictureUrl) 과 관련한 내용들을 업데이트한다.

```
CRSDKManager.getMyProfileForKakao(new CRCommonCodeListener()
{
    @Override
    public void onSuccessed()
    {
        CRUser user = CRSDKManager.getUser();
        CRSDKManager.pushLog("[setOnClickListener] getMyProfileForKakao is
succesded.");
    }

    @Override
    public void onFailed(int code)
    {
        CRSDKManager.pushLog("[setOnClickListener] getMyProfileForKakao is failed.");
    }
});
```


2.1.5 친구 리스트 요청 API

Kakao 친구 리스트를 가져오는 API 이다. 친구 리스트의 타입은 게임 미설치 친구(초대 가능 친구), 게임 설치 친구 등 두 가지 타입이 있다. 아래의 함수는 각각 초대 가능 친구 및 게임 설치 친구를 가져오는 API 들이다.

```
// 초대 가능 친구 리스트 가져오기(게임 미설치)
```

```
CRSDKManager.getInvitableFriendsForKakao(new CRKakaoFriendsResponseListener()
{
    @Override
    public void onSuccessed(List<FriendInfo> list)
    {
        if (list.size() <= 0) return;

        FriendInfo friend = list.get(0); // 첫 번째 친구를 가져온다.
        if (friend == null) return;
    }

    @Override
    public void onFailed()
    {
        CRSDKManager.pushLog("[setOnClickListener] getInvitableFriends is failed.");
    }
});
```

```
// 게임 설치 친구 가져오기
```

```
CRSDKManager.getRegisteredFriendsForKakao(new CRKakaoFriendsResponseListener()
{
    @Override
    public void onSuccessed(List<FriendInfo> list)
    {
        if (list.size() <= 0) return;

        FriendInfo friend = list.get(0); // 첫 번째 친구를 가져온다.
        if (friend == null) return;
    }

    @Override
    public void onFailed()
    {
        CRSDKManager.pushLog("[setOnClickListener] getInvitableFriends is failed.");
    }
});
```

2.1.6 초대 메시지 전송 API

게임 미설치(초대 가능) 친구에게는 게임을 설치하도록 유도하는 초대 메시지를 보낼 수 있다. 아래의 함수를 통하여 초대 메시지를 전송한다.

초대 메시지 전송을 위해서는 게임 센터 Admin 페이지에서 메시지 템플릿 설정이 되어 있어야 한다.

Admin/게임 프로젝트/메시지 템플릿 관리(Manage Message Template)에서 유효한 템플릿 아이디를 사업부 등에서 공유받아 사용하도록 한다.

```
// Parameter 1 : Message Template ID
// Parameter 2 : 친구 정보 객체(FriendInfo)
// Parameter 3 : CRCommonCodeListener 리스너

CRSDKManager.sendInviteMessageForKakao("2419", friend, new CRCommonCodeListener()
{
    @Override
    public void onSuccessed()
    {
    }

    @Override
    public void onFailed(int code)
    {
        CRSDKManager.pushLog("[sendInviteMessageForKakao]
sendInviteMessageForKakao is failed.");
    }
});
```

2.1.7 게임 메시지 전송 API

게임이 이미 설치된 친구에게는 게임 메시지를 전송할 수 있다.
아래의 함수를 통하여 게임 메시지를 전송하면 된다.

```
// Parameter 1 : Message Template ID
// Parameter 2 : 친구 정보 객체(FriendInfo)
// Parameter 3 : CRCommonCodeListener 리스너

CRSDKManager.sendGameMessageForKakao("2419", friend, new CRCommonCodeListener()
{
    @Override
    public void onSuccessed()
    {
    }

    @Override
    public void onFailed(int code)
    {
        CRSDKManager.pushLog("[sendInviteMessageForKakao]
sendInviteMessageForKakao is failed.");
    }
});
```

2.2 iOS API

VOLTUS SDK에서 제공되는 주요 iOS API를 설명한다.
좀 더 자세한 Kakao 관련 iOS API의 내용은 아래의 주소에서 확인바란다.

<https://gamecenter.kakao.com/docs/documents/288>

2.2.1 Login API

Kakao 로그인을 위해서는 아래와 함수를 호출한다.

기존의 VOLTUS SDK 의 didFinishedLogin, didFailedLogin 리스너 함수에서 로그인 여부를 확인할 수 있다.

```
[gCRSDKManager loginForKakao];
```

2.2.2 Logout API

Kakao 로그아웃을 위한 API이다.

기존의 VOLTUS SDK 의 didFinishedLogout, didFailedLogout 리스너 함수에서 로그인 여부를 확인할 수 있다.

```
[gCRSDKManager logoutForKakao];
```

2.2.3 로그인 확인(Verify) API

로그인을 한번 하면 일정 시간 동안에는 AccessToken이 유효하며 자동적으로 로그인을 할 수 있도록 된다.

따라서 게임 내에 로그인 관련 화면에서 로그인 버튼을 표시하기 이전에 현재 로그인 상태가 유효한지 체크하여 유효하면 자동적으로 로그인을 처리를 완료하여 로그인 이후 상태로 진입하면 된다.

아래의 함수는 현재의 로그인 상태를 알려주는 함수이다.

```
BOOL isSessionOpen = [gCRSDKManager verifyLoginForKakao];

if(isSessionOpen)
{
    CRSDKManager.pushLog("Login Session is already opened.");
}
```

2.2.4 카카오 프로필 API

카카오 관련 프로필을 가져올 수 있는 API 이다.

아래와 같이 호출하면 CRUser 내에 카카오 프로필(uid, name, pictureUrl) 과 관련한 내용들을 업데이트한다.

```
[gCRSDKManager getMyProfileForKakao:^(NSInteger code)
{
    CRUser *user = [gCRSDKManager getUser];
    if(user != nil)
    {
        }
    }
}];
```

2.2.5 친구 리스트 요청 API

Kakao 친구 리스트를 가져오는 API 이다. 친구 리스트의 타입은 게임 미설치 친구(초대 가능 친구), 게임 설치 친구 등 두 가지 타입이 있다. 아래의 함수는 각각 초대 가능 친구 및 게임 설치 친구를 가져오는 API 들이다.

```
// 초대 가능 친구 리스트 가져오기(게임 미설치)

[gCRSDKManager getInvitableFriendsForKakao:^(NSArray *list)
{
    if(list != nil && [list count] > 0)
    {
        KOFriend *friend = list[0]; // 첫 번째 친구를 가져온다.
        if(friend == nil) return;

        [CRLogManager pushLog:@"[getInvitableFriendsForKakao] nickname : %@",
friend.nickName];
    }
}];

// 게임 설치 친구 가져오기

[gCRSDKManager getRegisteredFriendsForKakao:^(NSArray *list)
{
    if(list != nil && [list count] > 0)
    {
        KOFriend *friend = list[0]; // 첫 번째 친구를 가져온다.
        if(friend == nil) return;
    }
}];
```

2.2.6 초대 메시지 전송 API

게임 미설치(초대 가능) 친구에게는 게임을 설치하도록 유도하는 초대 메시지를 보낼 수 있다. 아래의 함수를 통하여 초대 메시지를 전송한다.

초대 메시지 전송을 위해서는 게임 센터 Admin 페이지에서 메시지 템플릿 설정이 되어 있어야 한다. Admin/게임 프로젝트/메시지 템플릿 관리(Manage Message Template)에서 유효한 템플릿 아이디를 사업부 등에서 공유받아 사용하도록 한다.

```
// Parameter 1 : Message Template ID
// Parameter 2 : 친구 정보 객체(KOFriend)
// Parameter 3 : 코드값(0이면 성공)

[gCRSDKManager sendInviteMessageForKakao:@"2419" friendInfo:friend handler:^(NSInteger
code)
{
}];
```

2.2.7 게임 메시지 전송 API

게임이 이미 설치된 친구에게는 게임 메시지를 전송할 수 있다.
아래의 함수를 통하여 게임 메시지를 전송하면 된다.

```
// Parameter 1 : Message Template ID
// Parameter 2 : 친구 정보 객체(KOFriend)
// Parameter 3 : 코드값(0이면 성공)

[gCRSDKManager sendMessageForKakao:@"2419" friendInfo:friend handler:^(NSInteger code)
{
}];
```

3. 기타 유의사항

3.1 Kakao 스플래시 화면 적용

Kakao 스플래시 화면의 경우 최근에(2016.09) 동영상(2-3초) 방식으로 변경되었다.

스플래시 동영상에 경우에는 사용 게임 엔진 및 지원 해상도 등에 따라 상이하므로 아래의 가이드 및 디자인 Kit를 다운 받아 개발사에서 별도로 적용할 수 있도록 한다. 표시 시점에 대해서는 사업부의 정책에 따라 적용하면 된다.

Splash Image 가이드 : <https://gamecenter.kakao.com/docs/documents/172>
디자인 Kit 다운로드 : <https://gamecenter.kakao.com/docs/documents/169>(개발사에서의 지원 해상도와 방향에 따라서 상이하게 리소스를 적용하면 됨)

3.2 로그인 확인 프로세스

카카오 로그인 화면 등에 진입하면 반드시 2.1.3 로그인 확인(Verify) API를 먼저 호출 후 현재 세션이 Open 상태이면 카카오 로그인 버튼 표시 없이 바로 게임으로 진입하면 된다.

3.3 카카오 AccessToken 사용(Server to Server 적용 시 등)

카카오에서 제공하는 Server to Server 등의 API 적용 시에는 각 플랫폼 별로 아래와 같이 카카오 액세스 토큰이 필요 시 사용하면 된다.

카카오 액세스 토큰은 반드시 verifyLoginForKakao 함수의 반환 값이 참인 경우 내지는 로그인 성공 후에 호출하여야 정상적인 값을 얻을 수 있다.

```
Android : CRSDKManager.getAccessTokenKakao();
iOS : [gCRSDKManager accessTokenKakao];
```

3.4 페이스북 미지원(Android 만)

현재 제공되는 Kakao 지원용 Android SDK는 페이스북 등의 기능을 사용할 수 없다.
따라서 기존에 페이스북 lib 프로젝트(facebook-sdk)를 링크하거나 기능을 적용하였다면 제거할 수 있도록 한다.

3.5 카카오톡 미가입 유저 처리(Android 만)

카카오톡 미가입 유저란 카카오톡을 이메일 계정만 만들고 실제 휴대폰과 연동을 하지 않는 계정을 말한다. 해당 계정은 카카오톡 로그인만 가능하되 실제 카카오톡과 연동이 완료되지 않았으므로 프로필 정보 및 친구 정보가 없다. 해당 계정은 CRUser에서 isKakaoUser 값이 false로 설정되어진다. 또한 친구 초대 및 메시지 등의 카카오톡 관련 API 호출을 할 수 없다.