VOLTUS Client SDK for Android (Android Studio)

문서 개요

문서 개요

이 문서는 VOLTUS Client SDK 에 대한 소개 및 사용 방법에 대하여 가이드를 하는 문서이다.

대상 독자

이 문서의 대상은 VOLTUS Client SDK를 적용하려고 하는 개발사 및 개발자들이다. 해당 문서는 Android Studio 플랫폼을 위하여 가이드한다.

문서 이력

날짜	리비전	설명
2016.10.12	1.0	가이드 문서 최초 생성 및 배포
2016.11.15	1.1	**onAcitivyResult** 관련 가이드 수정
2016.11.17	1.2	**Android 5.0(lollipop, API 21)** 이상을 위한 별도의 푸시 아이콘 **ic_silhouette.png** 파일 가이드 추가
2016.11.24	1.3	권한 요청 팝업 API 추가
2017.02.23	1.4	Naver Cafe Plug SDK 연동 추가

SDK 버전 이력

버전 이력

날짜	버전	설명
2016.10.12	1.0.0	최초 Android Studio용 빌드 생성(기존 lib 0.9.9을 기반으로 생성)
2017.02.24	1.1.0	기존 lib 0.9.18 까지의 기능 적용, Naver Cafe Plug SDK 연동 추가

목차

목차

목차

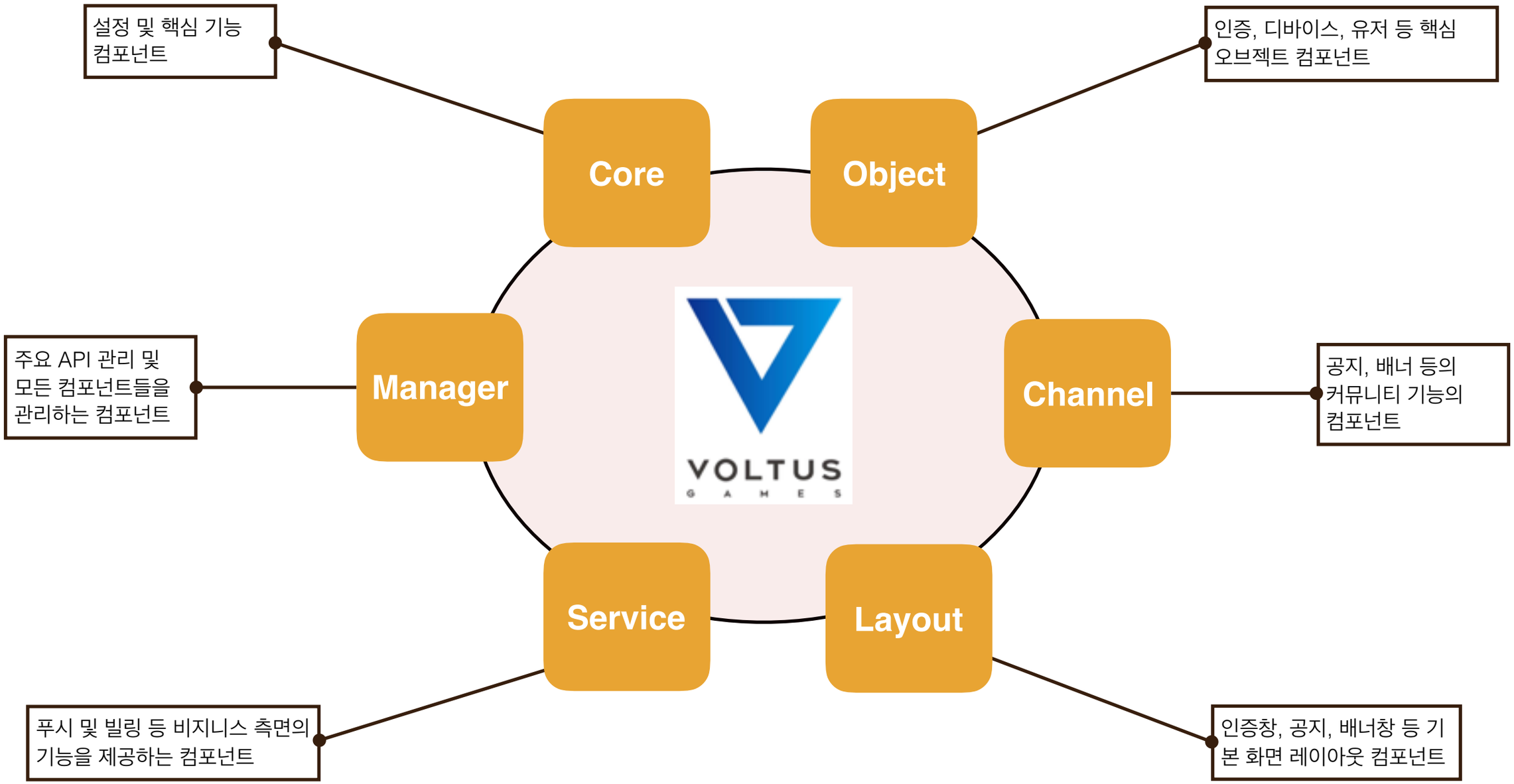
VOLTUS SDK 목적

VOLTUS SDK는 게임 개발 시 필요한 인증, 공지 등의 사용자 관리 및 운영 측면에서의 기능 들을 제공해주며 게임 개발사는 게임 컨텐츠 개발에만 집중할 수 있도록 하여 성공적인 게임 서비스 제공을 할 수 있도록 도와주는 플랫폼 서비스 지원 도구이다.

SDK 지원 서비스

서비스	설명
통합 소셜 사용자 인증	각종 SNS(Facebook, Google+ 등) 사용자들을 플랫폼 안에서 모두 통합하여 인증 및 사용자 관리 지원
공지 및 배너	게임 이벤트 및 공지, 각종 크로스 프로모션을 위하여 플랫폼 차원에서의 공지 및 배너 기능 지원
푸시 알림 서비스	iOS, Android 푸시 알림 기능을 지원하여 유저의 참여를 적극적으로 유도하도록 지원
통합 인앱 빌링	iOS, Android, OneStore 등 주요 인앱 기능을 하나의 인터페이스로 지원
쿠폰 서비스	각 게임에서의 쿠폰을 지원하여 다양한 형태의 이벤트 및 서비스를 할 수 있도록 지원
무료 충전소	VOLTUS 플랫폼의 게임 간의 크로스 프로모션 등을 위하여 자사의 타게임을 추가적으로 설치하면 보상 등을 지급하는 서비스
멤버십 서비스	플랫폼 자체의 멤버십 기능을 제공하여 유저들에게 각종 혜택 및 플랫폼 내의 모든 게임에 대한 혜택 지원(차후 지원 예정)

주요 컴포넌트 구성


설정 및 핵심 기능 컴포넌트
Core
Object
인증, 디바이스, 유저 등 핵심 오브젝트 컴포넌트
주요 API 관리 및 모든 컴포넌트들을 관리하는 컴포넌트
Manager
Channel
공지, 배너 등의 커뮤니티 기능의 컴포넌트
VOLTUS GAMES
Service
Layout
푸시 및 빌링 등 비지니스 측면의 기능을 제공하는 컴포넌트
인증창, 공지, 배너창 등 기본 화면 레이아웃 컴포넌트

Component : Core

Component	Class / File	기능
Core	CRConfig	ENUM 리스트 선언
Core	CRConstants	기타 설정값 및 매크로 선언
Core	CRNetwork	HTTP 통신을 위한 기능 지원
Core	CRUtil	각종 Utility 기능 지원

Component : Object

Component	Class / File	기능
Object	CRAuth	인증과 관련한 정보를 담고 있는 클래스
Object	CRDevice	장비(Device)와 관련한 정보를 담고 있는 클래스
Object	CRPlatform	플랫폼 정보(SNS 관련 정보)들을 담고 있는 클래스
Object	CRUser	유저에 대한 정보를 담고 있는 클래스

Component : Manager

Component	Class / File	기능
Manager	CRAPIManager	API 함수들을 관리하는 매니저 클래스
Manager	CRSDKManager	기본 SDK의 모든 기능들을 관리하는 매니저 클래스
Manager	CRPlatformManager	플랫폼(SNS)와 관련한 기능들을 관리하는 매니저 클래스
Manager	CRLogManager	로그 기록들을 관리하는 매니저 클래스

Component : Channel

Component	Class / File	기능
Channel	CRBanner	배너와 관련한 정보를 담고 있는 클래스
Channel	CRNotice	공지와 관련한 정보를 담고 있는 클래스

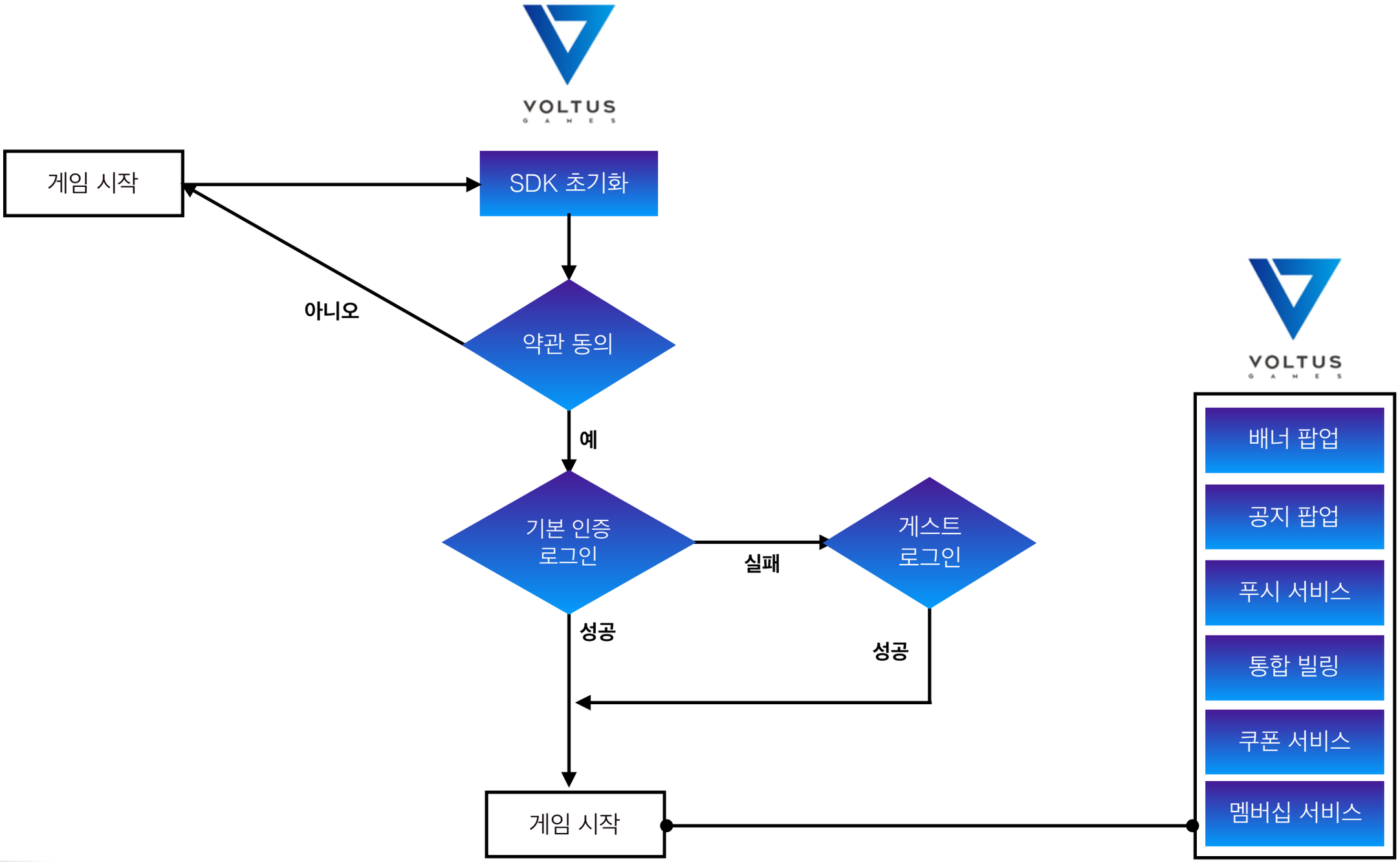
Component : Service

Component	Class / File	기능
Service	CRPush	푸시 설정 및 기능을 제공하는 클래스
Service	CRBilling	통합 빌링을 설정하고 기능을 제공하는 클래스

Component : Layout

Component	Class / File	기능
Layout	LoginActivity	소셜 로그인을 선택할 수 있도록 제공하는 레이아웃 클래스
Layout	LoginAuthActivity	소셜 인증을 처리하는 레이아웃 클래스
Layout	TermsActivity	약관을 표시하고 처리하는 레이아웃 클래스
Layout	CRWebActivity	공지를 위한 웹뷰 형식의 뷰 클래스
Layout	CRBannerActivity	배너 팝업을 위한 뷰 클래스

SDK 전반적인 흐름도


게임 시작
SDK 초기화
아니오
약관 동의
예
기본 인증
로그인
실패
게스트
로그인
성공
성공
게임 시작
배너 팝업
공지 팝업
푸시 서비스
통합 빌링
쿠폰 서비스
멤버십 서비스

기본 개발 요구 사항

Android Studios 2.x, Minimum Android SDK API 버전 15(Android 4.0.3) 이상

기본 기능 사용 요구 사항

VOLTUS 플랫폼 서버에 등록된 게임 ID
개발자 관련 스토어 기본 등록 정보들
Firebase 프로젝트 생성(FCM 사용시)

소셜 인증 및 기타 서비스 사용 시

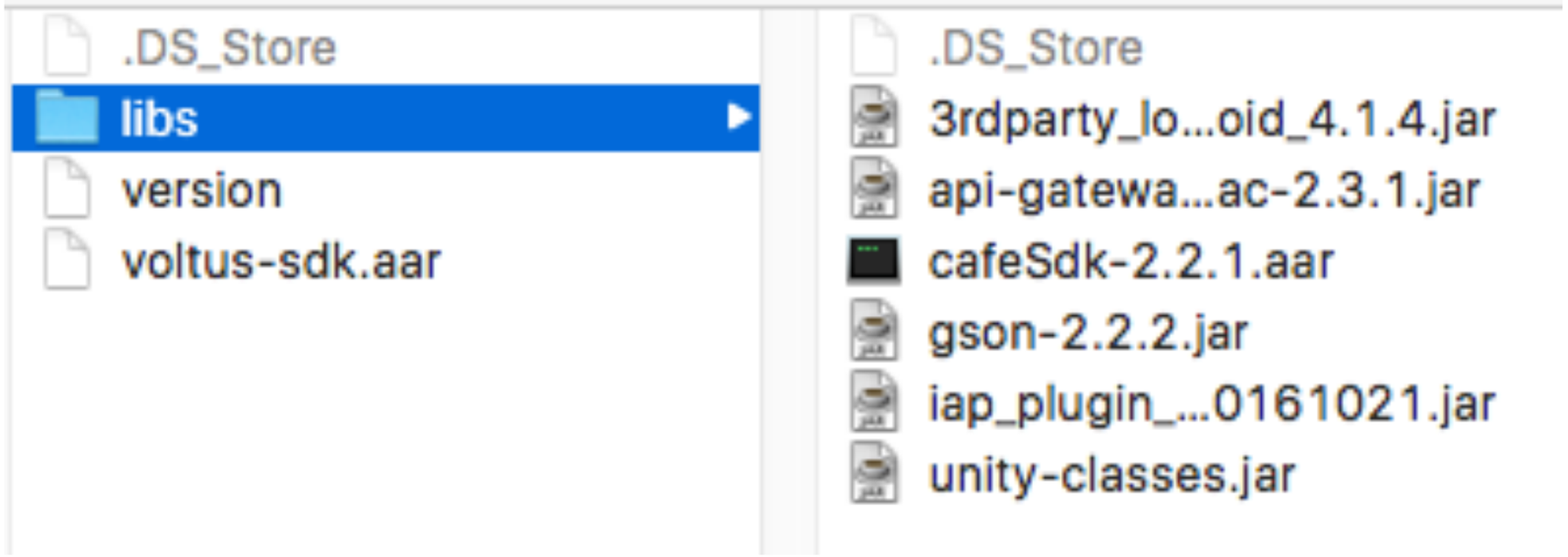
VOLTUS 서버와 연동할 서버 구축 및 API 사용, 서버 API는 서버 API 문서 참조

Facebook 인증 사용 시 : Facebook 기본 앱 생성(**https://developers.facebook.com/**), Facebook 개발자 페이지에서 릴리즈 키스토어의 키해시 등록
FCM 사용 시 : Firebase 콘솔에서 기본 앱 생성(**https://console.firebase.google.com**), 신규 프로젝트 생성 후 앱 설정,
기타 자세한 사항은 Firebase 관련 가이드 페이지(**https://firebase.google.com/docs/android/setup**) 에서 살펴보길 바란다.
만약 개발사에서 직접 Firebase 프로젝트를 직접 구성하지 않는 형태라면 큐로홀딩스 사업부 등에 문의하여 프로젝트 생성을 요청하면 된다.

SDK 관련 파일

SDK 관련 필요 파일들은 아래와 같다.

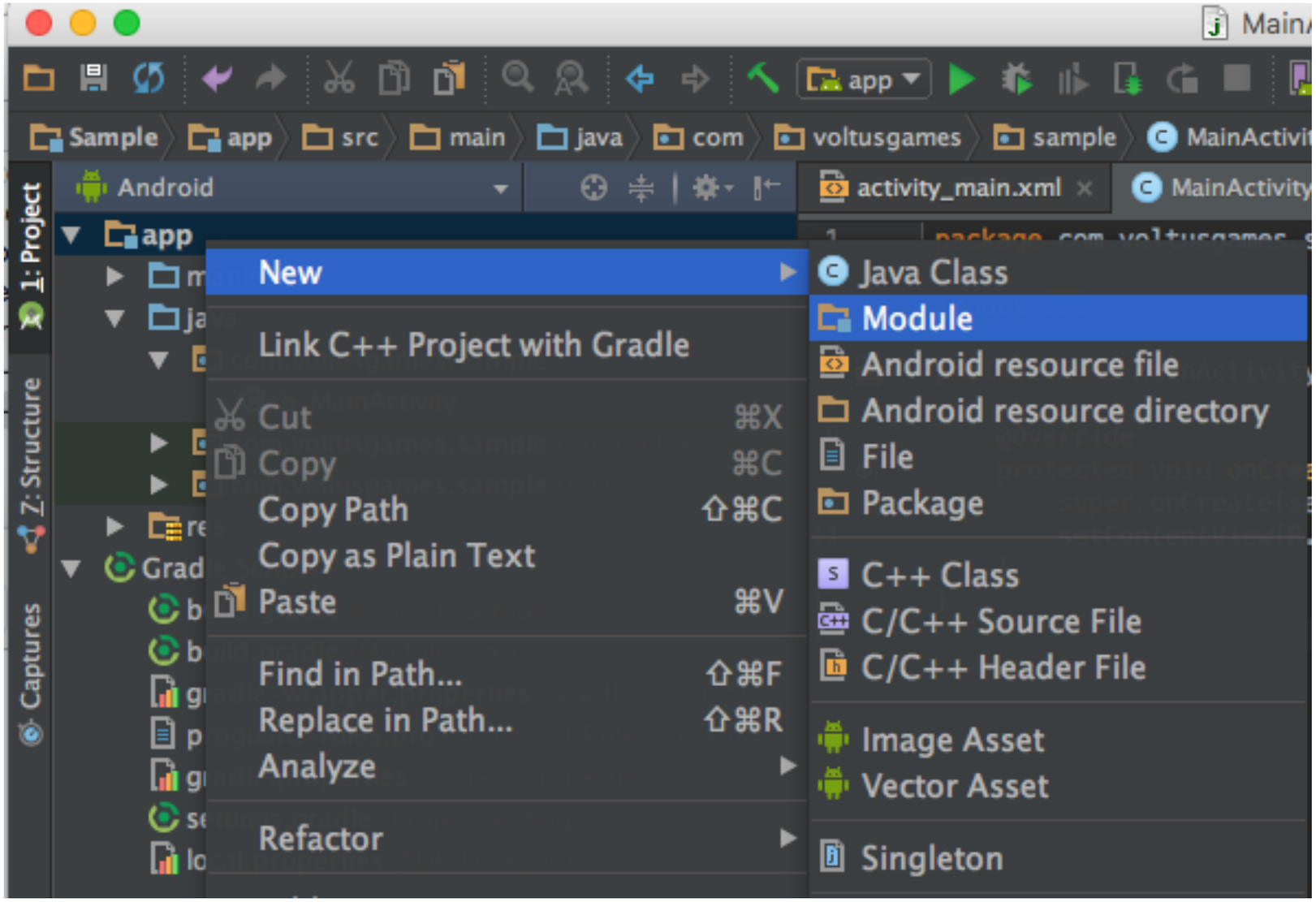
voltus-sdk.aar : 기본 Voltus SDK lib 파일
gson-2.2.2.jar : Voltus SDK에서 사용하는 기본 jar 파일
iap_plugin-15.01.00.jar : 원스토어 인앱을 위한 jar 파일
unity-classes.jar : Unity3d 프로젝트를 위한 jar 파일
3rdparty_login_library_android_4.1.4.jar : Naver Cafe Plug SDK 관련 jar 파일
api-gateway-hmac-2.3.1.jar : Naver Cafe Plug SDK 관련 jar 파일
cafeSdk-2.2.1.aar : Naver Cafe Plug SDK 관련 aar 파일



Android Studio 프로젝트 내에 VOLTUS SDK 모듈 임포트

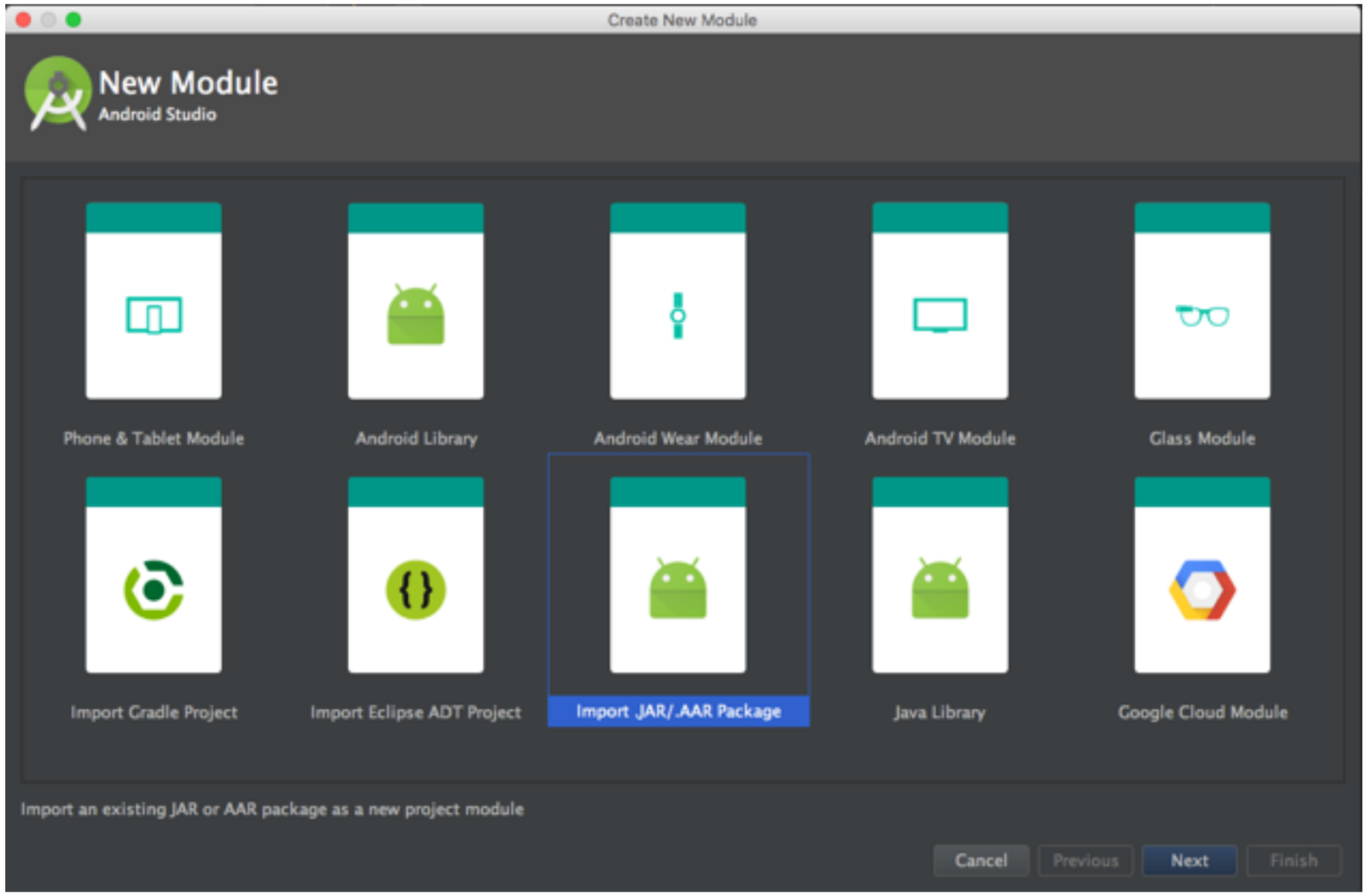
아래의 순서대로 Voltus SDK 기본 모듈들을 기본 게임 프로젝트에 임포트한다.

1) 기본 Root 프로젝트를 마우스 우측 버튼 선택 후 '**New -> Module**' 선택



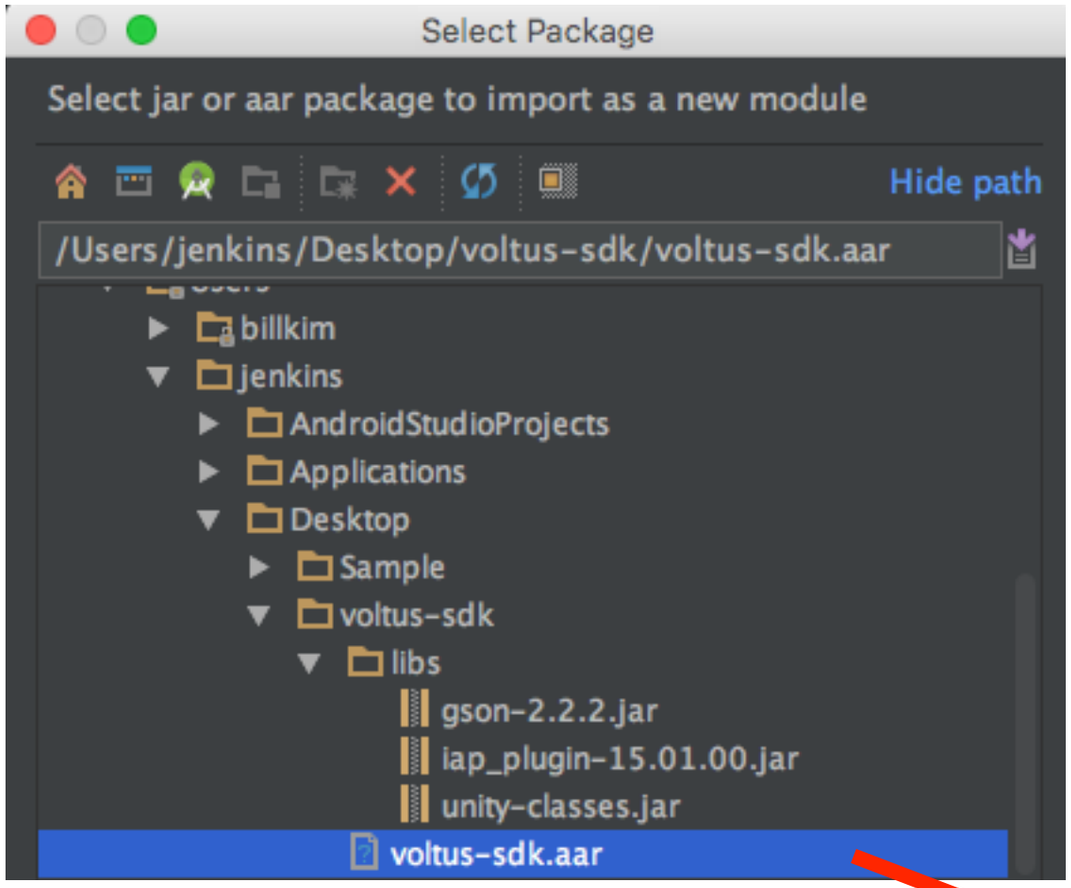
Android Studio 프로젝트 내에 VOLTUS SDK 모듈 임포트(계속)

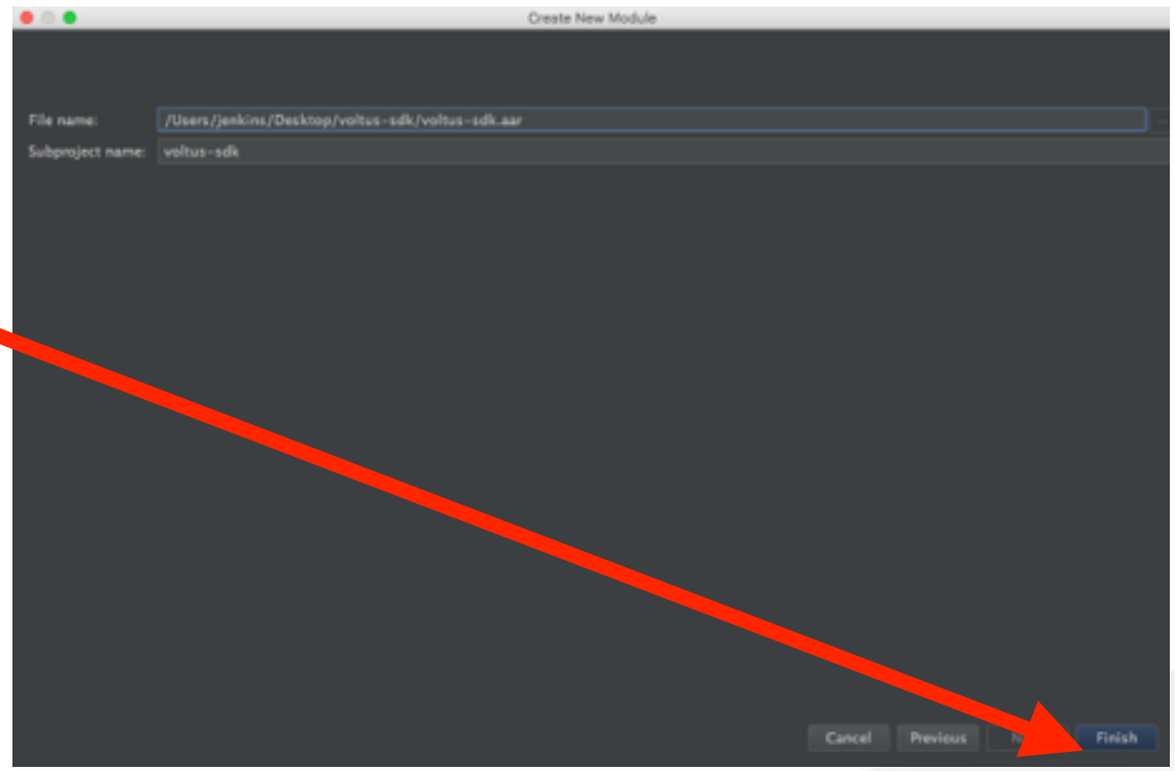
2) **Import .JAR/.AAR Package** 선택 후 Next 선택



Android Studio 프로젝트 내에 VOLTUS SDK 모듈 임포트(계속)

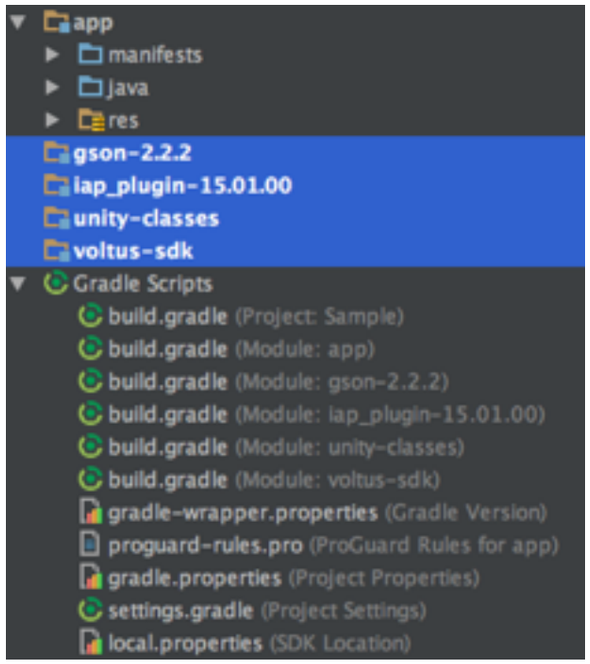
3) 각 모듈을을 **1) - 3)** 의 절차대로 각각 개별 수행**(voltus-sdk.aar, gson-2.2.2.jar, iap_plugin-15.01.00.jar, unity-classes.jar)**



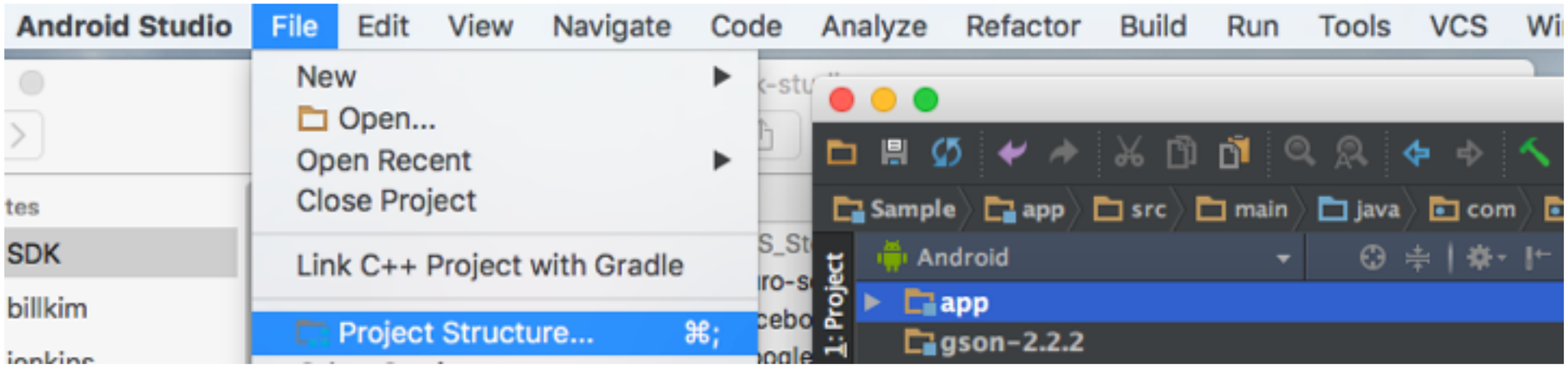


Android Studio 프로젝트 내에 VOLTUS SDK 모듈 임포트(계속)

4) 각 개별 모듈들을 모두 임포트 하면 아래와 같이 표시된다.

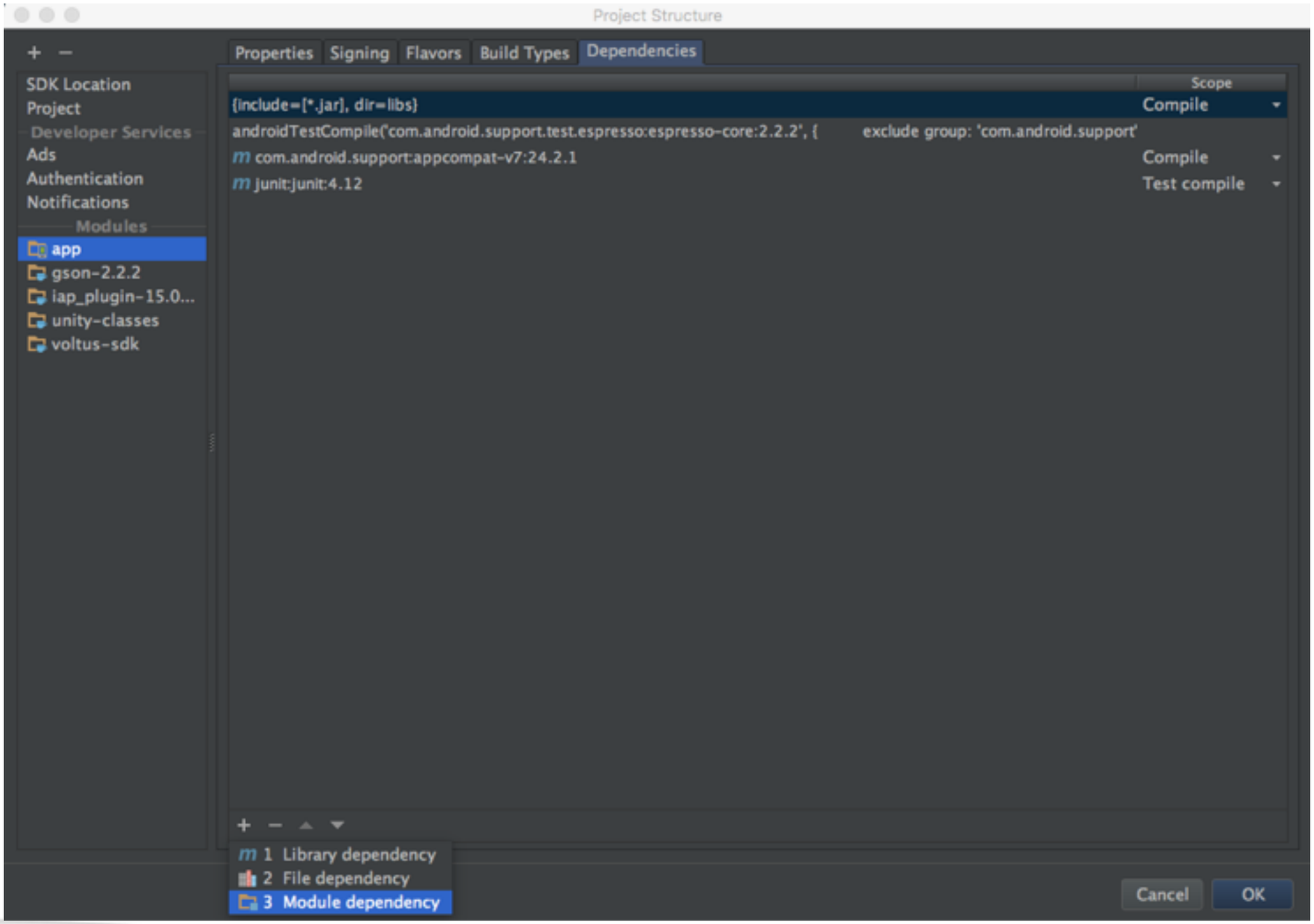


5) 기본 Root 프로젝트(여기서는 app)를 선택 후 ‘File -> Project Structure’를 선택한다.



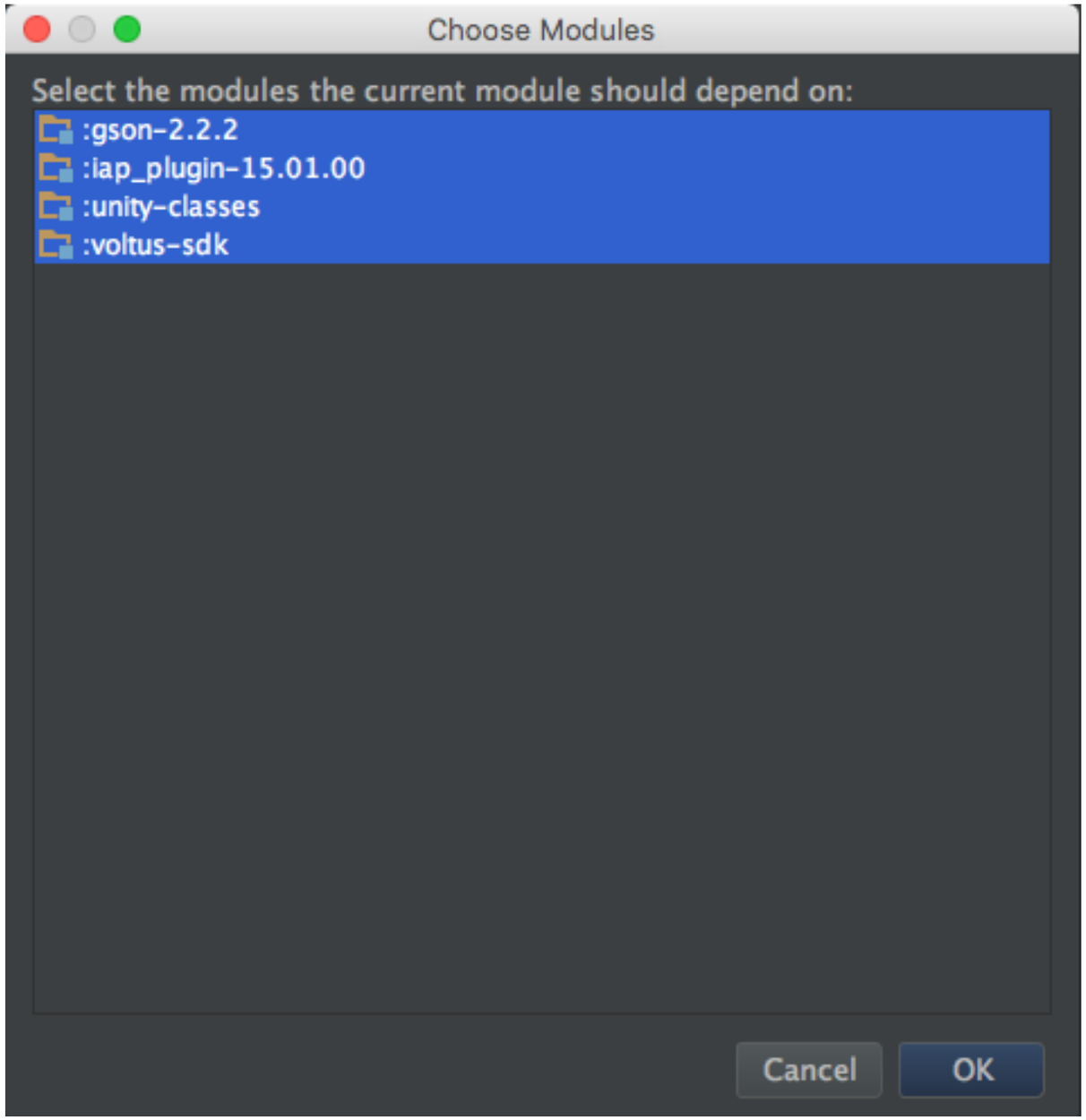
Android Studio 프로젝트 내에 VOLTUS SDK 모듈 임포트(계속)

6) Dependencies에서 하단 부 **'+' 버튼을 눌러 Module dependency**를 선택한다.



Android Studio 프로젝트 내에 VOLTUS SDK 모듈 임포트(계속)

7) 방금 전에 추가했던 4개의 모듈이 표시되면 다중으로 선택한 후 OK를 누른다.



Gradle Script 설정

Project build.gradle 설정(초록색으로 표시된 부분을 추가)

```
buildscript {
    repositories {
        jcenter()
    }

    dependencies {

        …

        classpath 'com.google.gms:google-services:3.0.0'

        …
    }
}

allprojects {
    repositories {

        …
        mavenCentral()
    }
}
```

Gradle Script 설정

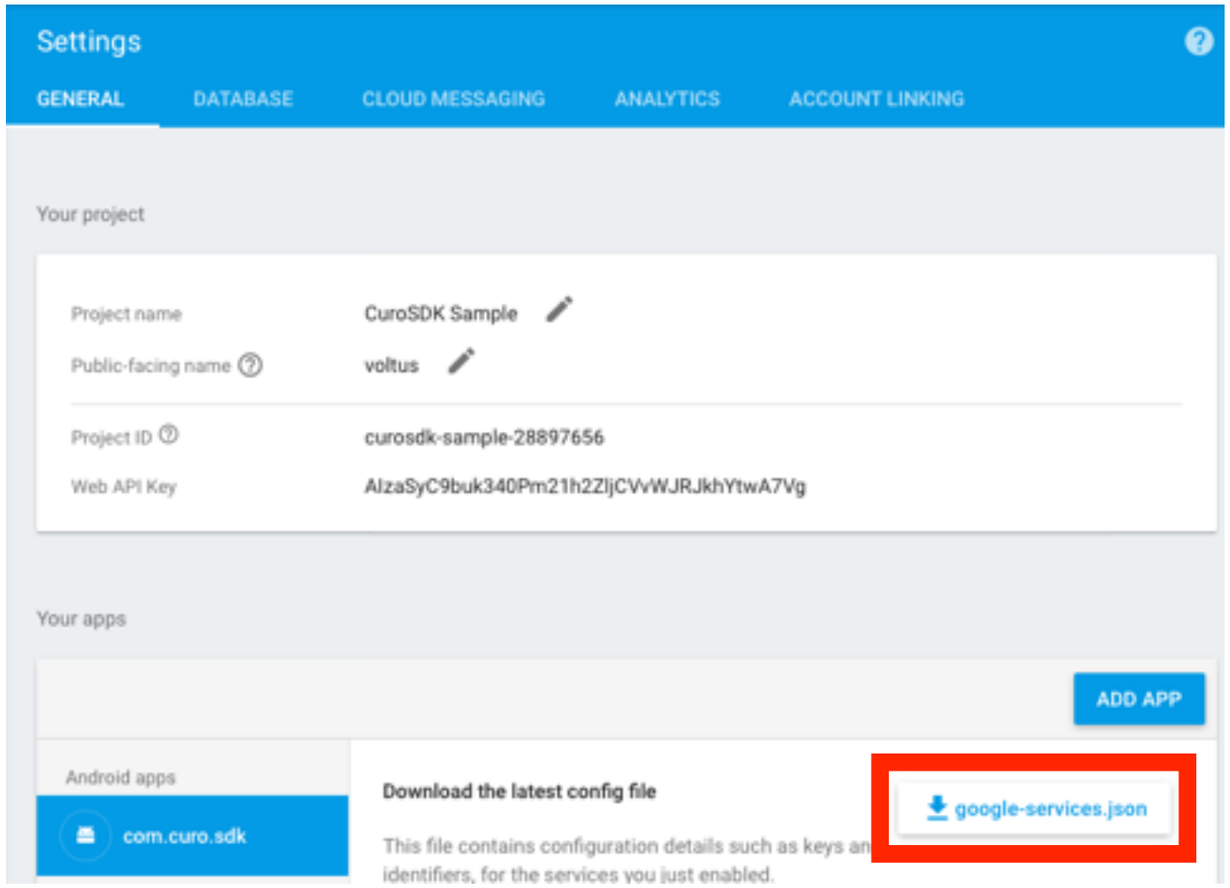
Module build.gradle 설정(초록색으로 표시된 부분을 추가)

```
android {
    defaultConfig {

        …

        multiDexEnabled  true

        …
    }

    …
}

dependencies {
    compile project(':voltus-sdk')
    compile project(':gson-2.2.2')
    compile project(':iap_plugin-15.01.00')
    compile project(':unity-classes')

    compile 'com.google.android.gms:play-services:9.6.1'
    compile 'com.google.firebase:firebase-messaging:9.6.1'

    compile 'com.facebook.android:facebook-android-sdk:4.+'
    compile 'com.facebook.android:audience-network-sdk:4.+'
}

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

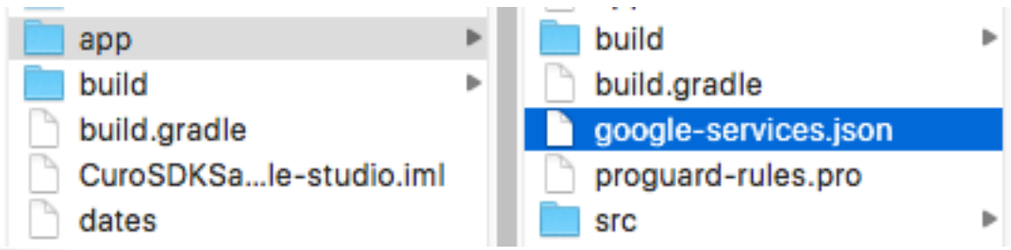
FCM 관련 google-service.json 파일 연결

1) Firebase 기본 콘솔에서 해당 게임 프로젝트의 Settings로 가면 아래와 같이 **google-service.json** 파일을 다운 받을 수 있다. 해당 파일을 다운받는다.

해당 관련 설정을 개발사가 직접 관리를 하지 않을 시 큐로홀딩스 사업부 등을 통하여 해당 파일(google-service.json)파일을 전달받는다.



2) 해당 파일(**google-service.json**)파일을 **프로젝트 기본 프로젝트 아래**에 옮긴다.



Permission 설정

필수 설정

AndroidManifest.xml 내에 기본 Permission 설정

```
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name=“android.permission.READ_EXTERNAL_STORAGE"/>
```

옵션 설정

구글 IAP(인앱) 이용 시

```
<uses-permission android:name="com.android.vending.BILLING"/>
```

원스토어 IAP(인앱) 이용 시

```
<uses-permission android:name="com.tmoney.vending.INBILLING"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

4) Layout Activity 설정

AndroidManifest.xml 내에 기본 Layout Activity 설정

```
<application
 ……

<activity
          android:name="com.curo.sdk.layout.CRLogoActivity"
          <!-- 방향을 고정하고 싶은 시에는 해당 설정을 activity 안에 방향과 함께 선언한다. android:screenOrientation="portrait" / (portrait,  landscape,
        sensorPortrait, sensorLandscape -->
          android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
          android:configChanges="keyboardHidden|orientation|screenSize" >
</activity>

<activity android:name="com.curo.sdk.layout.LoginActivity"
          android:theme="@android:style/Theme.NoTitleBar.Fullscreen">
</activity>

<activity android:name="com.curo.sdk.layout.LoginAuthActivity"
          android:theme="@android:style/Theme.NoTitleBar.Fullscreen">
</activity>

<activity android:name="com.curo.sdk.layout.CRWebviewActivity"
          android:theme="@android:style/Theme.NoTitleBar.Fullscreen">
</activity>

<activity android:name=“com.curo.sdk.layout.PushPopupActivity"
          android:screenOrientation=“portrait" <!-- PushPopupActivity의 경우는 플랫폼 정책으로 고정이므로 반드시 portrait로 설정하여야 한다. -->
          android:configChanges="keyboardHidden|orientation|screenSize">
</activity>

<activity android:name="com.curo.sdk.layout.TermsActivity"
          android:theme="@android:style/Theme.NoTitleBar.Fullscreen">
</activity>

<activity android:name="com.curo.sdk.layout.CRBannerActivity"
          android:theme="@android:style/Theme.Translucent">
</activity>

 ……..
</application>
```

5) Facebook SDK 이용시(옵션)

AndroidManifest.xml 내에 Facebook 관련 설정값 지정

```
<application
 ……..

<activity android:name="com.facebook.FacebookActivity"> </activity>

<meta-data
            android:name="com.facebook.sdk.ApplicationId"
            android:value="@string/fb_app_id"/>
 ……..
</application>
```

string.xml 내에 fb_app_id 설정

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="fb_app_id">1717720575134576</string>
</resources>
```

SDK 초기화 함수 선언

메인 엑티비티 내에서 **OnCreate** 또는 **onFinishedShowLogo** 리스너 호출 이후에 아래와 같은 SDK 초기화 함수를 추가한다.
가급적 **onFinishedShowLogo** 리스너 함수 호출 후에 초기화를 하도록 권장한다.
현재 빌드가 원스토어 빌드인 경우에만 두 번째 파라미터인 '원스토어 게임 아이디'를 입력하고 그렇지 않을 경우에는 반드시 빈값("")을 넣도록 한다.

```
CRSDKManager.initWithGameID("10000", // VOLTUS 플랫폼 게임 아이디
                            "OA00701082", // 원스토어 게임 아이디
                            CRConfig.CRServiceMode.CRServiceModeBeta, // 서비스 모드
                            mContext, // Main Context
                            "URL"); // 무료 충전소 관련 아이템 지급 서버 URL
```

```
@Override
public void onStart()
{
	super.onStart();

	CRSDKManager.onCRStart(this);
}

@Override
public void onDestroy()
{
	super.onDestroy();

	CRSDKManager.onCRDestroy(this);
}

@Override
public void onResume()
{
	super.onResume();

	CRSDKManager.onCRResume(this);
}

@Override
public void onPause()
{
	super.onPause();

	CRSDKManager.onCRPause(this);
}
```

onActivityResult 함수 부분에 아래와 같이 코드를 추가한다. onActivityResult 함수가 Override가 되어 있지 않을 시에는 추가하여 준다.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    if(!CRSDKManager.onActivityResult(requestCode, resultCode, data))
    {
       super.onActivityResult(requestCode, resultCode, data);
    }
}
```

메인 리스너 선언

아래와 같은 메인 리스너를 초기화 함수 등에서 추가하여 준다.

```
CRSDKManager.setMainListener(new CRMainListener()
{
        @Override
        public void onFinishedShowLogo()
        {
        }

        @Override
        public void onFinishedLogin()
        {
        }

        @Override
        public void onFailedLogin(int code)
        {
        }

        @Override
        public void onFinishedLogout()
        {
        }

        @Override
        public void onFailedLogout()
        {
        }

        @Override
        public void onReceiveLinkType(String type)
        {
        }

        @Override
        public void onReceivePushUrl(String uri)
        {
        }

        @Override
        public void onAgreedTerms()
        {
        }

        @Override
        public void onFinishedBilling(int state)
        {
        }

        @Override
        public void onReceivedResponseCode(int code)
        {
        }

        @Override
        public void onFinishedBannerPopup(String type)
        {
        }

        @Override
        public void onAllowedPermissions()
        {
        }
});
```

메인 리스너 목록

리스너 메소드	설명
onFinishedShowLogo	플랫폼 로고가 등장하고 끝났을 때 호출되는 리스너 함수
onFinishedLogin	로그인 성공 시에 호출되는 리스너 함수
onFailedLogin	로그인 실패 시에 호출되는 리스너 함수(자세한 코드에 대해서는 **5.3 코드 목록표** 참조. 33페이지)
onAgreedTerms	약관을 모두 동의할 시에 호출되는 리스너 함수
onFinishedLogout	로그아웃 성공 시에 호출되는 리스너 함수
onFailedLogout	로그아웃 실패 시에 호출되는 리스너 함수
onNotConnectedNetwork	네트워크 연결 실패 시에 호출되는 리스너 함수
onReceiveLinkType	배너 팝업에서 메인 이미지를 터치 시에 호출되는 리스너 함수
onReceivePushUrl	푸시 수신 시에 app 타입의 액션이 수신 시 호출되는 리스너 함수, web 타입은 호출되지 않는다.
onFinishedBilling	인앱 구매 성공 및 실패 여부를 알려주는 리스너 함수(**5.3 코드 목록표** 참조)
onReceivedResponseCode	쿠폰 등록 및 기타 코드 값을 수신 시 호출되는 리스너 함수(CRResponseCode 참조, **5.3 코드 목록표** 참조)
onFinishedBannerPopup	모든 배너 출력이 완료되면 호출되는 리스너 함수(**5.3 코드 목록표** 참조)
onAllowedPermissions	모든 권한 요청에 대하여 모두 수락하였을 경우 호출되는 리스너 함수

소셜 인증 기능

VOLTUS 플랫폼은 각종 소셜 인증 기능을 지원한다. 버전 별 소셜 인증 기능 지원 목록은 아래와 같다.
Android 플랫폼에서는 기본적으로 GPGS(Google Play Game Service)를 기본 소셜 인증으로 사용하여 GPGS 만 플랫폼 로그인을 시도하며, GPGS외의 플랫폼 타입은 외부 소셜(3rd Party) 로그인 및 기능 사용에만 활용하며 로그인 성공 후에는 기본적으로 플랫폼 로그인은 시도하지 않는다.

버전	지원 소셜 인증
1.0.0	Facebook, Google+, **GPGS(Google Play Game Service)**

로그인 팝업 API(단순 소셜 로그인)

해당 함수 호출 시 플랫폼에서 기본 지원하는 로그인 선택 화면이 나오고 해당 버튼 터치 시 해당 소셜에 대한 인증 과정을 진행한다.

```
CRSDKManager.showLoginPopup(mContext);
```

로그인 API(별도 팝업 출력 X, 소셜 로그인만 시도)

해당 함수 호출 시 원하는 플랫폼에 대해서 별도의 팝업없이 로그인 과정을 진행한다. **GPGS 로그인 시에는 CRPlatformTypeGooglePlay 으로 로그인을 시도한다.**

```
CRSDKManager.login(CRConfig.CRPlatformTypeGooglePlay, mContext);
```

Parameters

```
public interface CRPlatformType
{
      int CRPlatformTypeNone = -1;
      int CRPlatformTypeFacebook = 0;
       int CRPlatformTypeGooglePlus = 1;
       int CRPlatformTypeGooglePlay = 101;
}
```

Context

연결할 메인 엑티비티의 Context

로그아웃 API

해당 함수 호출 시 기존 로그인 되어 있던 정보를 모두 초기화하며 해당 플랫폼에서 로그 아웃을 한다.

```
CRSDKManager.logout();
```

로그인 상태 체크 API

현재 소셜 로그인이 되어 있는지 확인하는 함수이다. 이 함수는 SDK 내부적으로 앱 실행 시작 시 체크한다. 따라서 특수한 상황을 제외하고는 별도로 호출할 필요는 없다.

```
CRSDKManager.verifyLogin();
```

Return

```
boolean

true : 로그인 되어 있을 경우
false : 로그인이 되어 있지 않을 경우
```

계정 전환 방법

Android 플랫폼에서는 **기존 GPGS 계정을 전환**할 수 있는 기능이 가능하다. 계정 전환을 위한 프로세스는 아래와 같이 진행하면 된다.

1. **CRSDKManager.logout() 함수를 호출**하여 기존의 계정을 로그아웃 한다.
2. 정상적으로 로그아웃이 되어 **onFinishedLogout 리스너 함수가 호출**되면 **CRSDKManager.login(CRConfig.CRPlatformTypeGooglePlay, mContext);** 함수를 호출하여 GPGS를 재로그인을 시도한다.
3. **GPGS 로그인을 시도하면 Google 계정 선택 창이 나온다.** 유저가 원하는 계정을 눌러서 해당 계정으로 전환하여 로그인을 하면 된다.
4. **onFinishedLogin** 리스너 함수가 호출되어 **GPGS 로그인이 성공**하면 **유저에게 계정 연동 진행 확인 팝업을 조건에 따라 개발사가 자체적으로 띄운다.**
   만약 계정 연동을 하지 않기로 유저가 선택하면 반드시 **CRSDKManager.logout()** 을 하여 다시 로그아웃 상태로 만든다.

약관 팝업 출력 API

해당 함수 호출 시 기본 플랫폼 약관 팝업을 출력한다.

```
CRSDKManager.showTermsPopup(mContext, true);
```

Parameters

Context

연결할 메인 엑티비티의 Context

boolean

true : 강제로 팝업 출력
false : 이미 약관 동의 시 팝업 미출력

배너 팝업 출력 API

해당 함수 호출 시 기본 플랫폼 약관 팝업을 출력한다.
해당 함수는 호출 시점 등을 사업부 등과 협의하여 필요한 시점에 호출하도록 한다.

```
CRSDKManager.showBannerPopup(mContext, CRConstants.BANNER_TYPE_LOGIN);
```

Parameters

Context

연결할 메인 엑티비티의 Context

String

CRConstants.BANNER_TYPE_LOGIN : 로그인 팝업
CRConstants.BANNER_TYPE_EXIT : 게임 종료 팝업

공지 팝업 출력 API

해당 함수 호출 시 기본 플랫폼 약관 팝업을 출력한다.
해당 함수는 호출 시점 등을 사업부 등과 협의하여 필요한 시점에 호출하도록 한다. 공지 팝업 내에는 '**무료 충전소**' 기능이 포함되어 있다.

```
CRSDKManager.showNoticeListPopup(mContext, false);
```

Parameters

Context

연결할 메인 엑티비티의 Context

boolean

true : 강제로 팝업 출력
fales : 하루 보지 않기를 선택했다면 미출력

푸시 알림을 위한 기본 설정

VOLTUS SDK는 기본적인 푸시 서비스를 위하여 **FCM(Firebase Cloud Messaging)**을 사용한다.

정상적인 푸시 수신을 위해서는 AndroidManifest.xml 내에 아래의 service 태그를 입력하여야 한다.

```
<application

 ……

<service
    android:name="com.curo.sdk.service.CRFCMListenerService">
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT" />
    </intent-filter>
</service>

    .

<service
    android:name="com.curo.sdk.service.CRFCMInstanceIDListenerService">
    <intent-filter>
        <action android:name="com.google.firebase.INSTANCE_ID_EVENT" />
    </intent-filter>
</service>
 ……..

</application>
```

푸시를 수신 시에 표시될 아이콘 이미지는 메인 프로젝트에서 res/mipmap-xx 등의 폴더를 생성하고 파일명을 **ic_launcher.png** 으로 지정하여 프로젝트 내에 포함하면 되고 다중 해상도를 위해서는 각 해상도별(mipmap-hdpi, mipmap-ldpi, mipmap-mdpi, mipmap-xhdpi, mipmap-xxhdpi 등)로 해상도에 맞는 이미지를 같은 파일명으로 포함시키면 된다. TargetSDKVersion이 Android 5.0(lollipop, API 21) 이상을 위하여 추가적으로 ic_silhouette.png 파일명으로 투명값을 지원하는 푸시 아이콘 파일이 별도로 존재하여야 한다.

푸시 알림 초기화 API

해당 함수 호출 시 푸시 설정과 관련하여 초기화 작업을 진행한다.

```
CRSDKManager.initPush(mContext);
```

푸시 알림으로 앱 실행 시 API

푸시 알림 수신 시 액션 타입이 “app”으로 설정 시에는 푸시 알림 터치로 앱 기동 시에는 해당 리스너 함수가 호출된다. 특정 스키마 uri에 대하여 특정 처리가 필요할 시에는 아래 델리게이트 함수를 받아 동작을 지정하여 주면 된다.

예) 특수 이벤트 인앱 상품 프로모션 푸시 알림 수신 후 -> 해당 상품 페이지로 게임 내에서 이동되도록 구현

```
@Override
public void onReceivePushUrl(String uri)
{

}
```

구글 게임 서비스(GPGS) 초기 설정

AndroidManifest.xml 내에 Google Play Service 관련 설정값 지정

```
<application
 ……

<meta-data android:name="com.google.android.gms.games.APP_ID" android:value="@string/app_id"/>
<meta-data android:name="com.google.android.gms.appstate.APP_ID" android:value="@string/app_id"/>
<meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version"/>
 ……..
</application>
```

string.xml 내에 app_id 설정

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_id">265851265354</string>
</resources>
```

구글 게임 서비스(GPGS) 로그인 API

Android 플랫폼의 경우에는 구글 플레이 게임 서비스(GPGS)가 기본 인증으로 사용됨으로 필수이다.
GPGS 로그인은 앞서 설명한 로그인 API를 통하여 시도하면 되며 GPGS 로그인이 성공하였으면 자동적으로 **onFinishedLogin**이 호출되며 CRUser 값이 설정된다.

```
CRSDKManager.login(CRConfig.CRPlatformTypeGooglePlay, mContext);
```

GPGS 업적 관련 API

업적 달성 팝업 오픈과 업적을 오픈하는 함수는 아래와 같이 사용하면 된다.

```
// 업적 달성 팝업 출력
CRSDKManager.openAchievementUI();

// 업적 오픈 – 업적 아이디를 입력하면 해당 업적을 오픈할 수 있다.
CRSDKManager.unlockAchievement("업적아이디");
```

GPGS 리더보드 관련 API

업적 달성 팝업 오픈과 업적을 오픈하는 함수는 아래와 같이 사용하면 된다.

```
// 리더 보드 팝업을 출력한다.
CRSDKManager.openLeaderboardUI();

// 리더 보드 점수 등록
CRSDKManager.submitScoreToLeaderboard("리더보드 아이디", "점수");
```

IAP(인앱) 구매 API

해당 함수를 호출하여 특정 IAP(인앱) 상품을 구매한다.
해당 함수 호출 후에는 게임 내에서 사용하는 대기 화면 등을 출력할 것을 권장하며, onFinishedBilling 이벤트 리스너 수신 후 대기 화면을 종료한다.

```
CRSDKManager.purchaseItem("스토어 타입", "상품아이디", "플랫폼 서버에서 받은 TXID", "아이템 지급요청을 호출할 서비스 서버 경로");
```

Parameters

String storeType

: 스토어 타입(play : 구글 플레이 스토어, one : 원스토어)

String productID

: 스토어에 등록된 상품 아이디

String txID

: 플랫폼 서버에서 구매 예약 시 발급받은 txID

String itemUrl

: 아이템 지급 요청을 호출할 서비스 서버 경로 URL

원스토어 IAP 기능을 사용 시에는 아래와 같이 AndroidManifest.xml 파일안에 아래의 permission과 추가적인 meta-data를 선언하여 준다.

```
<uses-permission android:name="com.tmoney.vending.INBILLING" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
```

```
<application
 ……
      <meta-data
            android:name="iap:api_version"
            android:value="3" />

      <meta-data
            android:name="iap_app_id"
            android:value="OA00701082" /> <—— 이 부분에는 원스토어의 앱 아이디를 입력한다.

      <activity
            android:name="com.skplanet.dodo.IapWeb"
            android:configChanges="orientation|screenSize|locale|keyboardHidden|layoutDirection"
            android:excludeFromRecents="true"
            android:windowSoftInputMode="stateHidden" >
      </activity>
 ……..
</application>
```

플랫폼 로고 출력 API

해당 함수를 호출하여 플랫폼 로고를 출력한다. 보통 앱 시작 처음에 해당 로고를 보여준다. 플랫폼 로고 표시 완료 후 **onFinishedShowLogo** 리스너 함수가 호출되며, 해당 리스너 함수가 호출되면 플랫폼 초기화 코드 **CRSDKManager.initWithGameID** 함수를 호출하는 것을 권장한다.

```
CRSDKManager.showLogoPopupm(Context);
```

Parameters	
Context	연결할 메인 엑티비티의 Context

쿠폰 등록 API

해당 함수를 호출하여 쿠폰을 등록을 요청한다.
등록 요청 후 onReceivedResponseCode 이벤트 리스너를 통하여 쿠폰 등록에 관한 코드를 받아온다. 코드 값을 0을 수신 시 정상 등록이 된 경우이다.

```
CRSDKManager.registerCoupon("쿠폰 코드", "쿠폰 등록 요청을 호출할 서비스 서버 경로");
```

Parameters

String couponCode

: 등록 요청할 쿠폰 코드

String itemUrl

: 쿠폰 등록 요청을 호출할 서비스 서버 경로 URL

무료 충전소 팝업 출력 API

무료 충전소는 기본적으로 공지 사항 팝업 안에 포함되어 있다.
하지만 **별도로 무료 충전소 팝업을 띄우고 싶은 경우**에는 아래의 함수를 호출하여 무료 충전소 화면을 출력할 수 있다.

```
CRSDKManager.showFreeChargingPopup();
```

무료 충전소 아이템 서버 설정 API

기본적으로 아이템 지급 서버 설정은 **CRSDKManager.initWithGameID()** 함수 안에 파라미터(itemUrl)로 입력하게 되어 있으며, 초기화 함수 이후에 변경을 원하면 아래의 함수를 통하여 아이템 지급을 위한 서버의 URL을 변경하면 된다.

```
CRSDKManager.setItemServerURL("아이템 지급 서버 URL");
```

Parameters

`String url`

: 아이템을 지급하는 서버의 URL 주소

푸시 수신 동의 설정 API

푸시 수신 동의 여부를 정할 수 있는 API 함수이다. 해당 함수로 true로 하면 푸시 수신을 동의하게 되며, false로 하게 되면 수신을 거부하여 해당 유저에게는 푸시가 전송이 되지 않는다. 해당 함수 호출 후 **onSuccessed()** 함수가 호출되면 수정이 성공한 경우이다.

```
CRSDKManager.modifyPushPermission(true, new CRCommonListener()
{
   @Override
   public void onSuccessed()
   {
       CRSDKManager.pushLog("modifyPushPermission is successed.");
   }

   @Override
   public void onFailed()
   {
       CRSDKManager.pushLog("modifyPushPermission is failed.");
   }
});
```

Parameters

`boolean allow`

: 푸시 수신 동의 여부. 수신 동의 시 true

야간 푸시 수신 동의 설정 API

야간 푸시 수신 동의 여부를 정할 수 있는 API 함수이다. 해당 함수로 true로 하면 야간 푸시 수신을 동의하게 되며, false로 하게 되면 수신을 거부하여 해당 유저에게는 야간 푸시가 전송이 되지 않는다. 해당 함수 호출 후 **onSuccessed()** 함수가 호출되면 수정이 성공한 경우이다.

```
CRSDKManager.modifyNightPushPermission(true, new CRCommonListener()
{
   @Override
   public void onSuccessed()
   {
       CRSDKManager.pushLog("modifyNightPushPermission is successed.");
   }

   @Override
   public void onFailed()
   {
       CRSDKManager.pushLog("modifyNightPushPermission is failed.");
   }
});
```

Parameters

`boolean allow`

: 야간 푸시 수신 동의 여부. 수신 동의 시 true

권한 요청 팝업 API

권한 요청에 대한 수락을 받기위한 함수로서 아래와 같은 방식으로 원하는 시점에 호출하여 준다.
해당 함수는 권한 사용에 대한 사유 및 각 권한에 대한 수락을 요청하는 팝업을 출력시킨다. 만약 모든 권한을 수락하지 않으면 게임에 진입할 수 없게 된다.

```
CRSDKManager.checkPermissions();
```

MainActivity 내에 코드 추가

권한 요청 팝업 API 사용시에는 게임에서 메인으로 사용하는 MainActivity 내에 아래의 코드를 적용하여야 한다.

```
public class MainActivity extends Activity implements ActivityCompat.OnRequestPermissionsResultCallback
{
..

     public void onRequestPermissionsResult(int requestCode,
                                            @NonNull String[] permissions,
                                            @NonNull int[] grantResults)
      {
            CRSDKManager.isAllowedAllPermissions(grantResults);
      }

..
}
```

모든 권한 수락 시의 리스너 함수

CRSDKManager.checkPermissions() 함수를 호출하면 현재의 권한 설정 허용 여부를 확인하게 되며 모든 권한 수락 시에는 아래의 리스너 함수가 호출되게 된다.

```
@Override
public void onAllowedPermissions()
{

}
```

Naver Cafe Plug SDK 연동

Naver Cafe PLUG SDK는 Naver에서 제작한 모바일 게임에서 이탈하지 않고 커뮤니케이션이 가능한 IN-GAME COMMUNITY 라이브러리이다.
Naver Cafe PLUG SDK는 손쉽게 커뮤니케이션 기능을 게임 안에 넣을 수 있다.

AndroidManifest 설정

동영상 녹화 기능을 사용할 때 필요한 접근 권한을 추가한다.
<uses-permission android:name="android.permission.RECORD_AUDIO"/>

Activity 추가

```
<activity
        android:name="com.nhn.android.naverlogin.ui.OAuthLoginActivity"
        android:screenOrientation="sensorLandscape"
        android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
    <activity
        android:name="com.nhn.android.naverlogin.ui.OAuthLoginInAppBrowserActivity"
        android:screenOrientation="sensorLandscape"
        android:label="OAuth2.0 In-app"/>
    <activity
       android:name="com.naver.glink.android.sdk.ui.VideoPlayActivity"
       android:screenOrientation="sensorLandscape"
       android:theme="“@android:style/Theme.Translucent.NoTitleBar.Fullscreen"/>
```

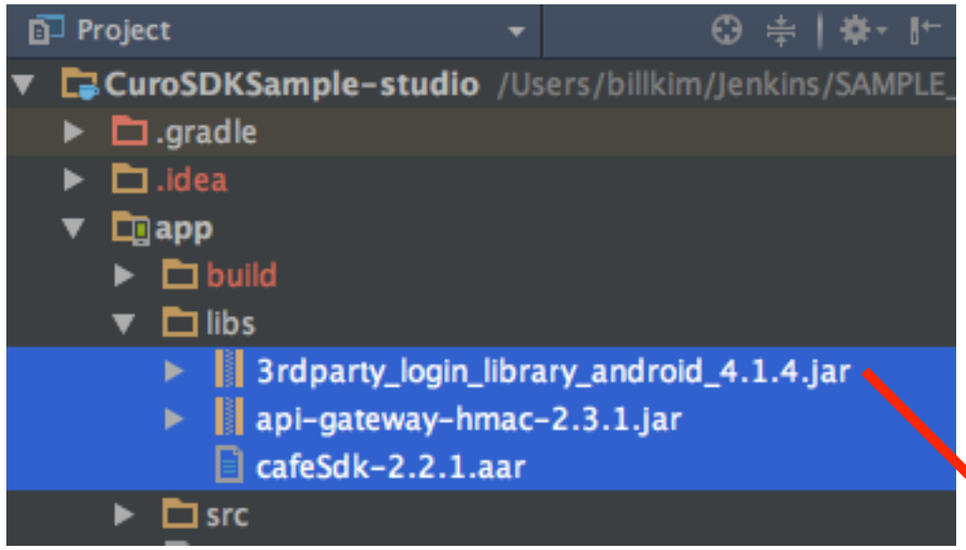
글로벌 카페를 사용할 경우 아래의 NeoIdAppBrowserActivity가 추가되어야 한다.

```
<activity
        android:name="com.naver.glink.android.sdk.login.neoid.NeoIdInAppBrowserActivity"
        android:configChanges="keyboardHidden|screenSize|orientation"
        android:label="NeoId In-app"
        android:screenOrientation="sensor"/>
```

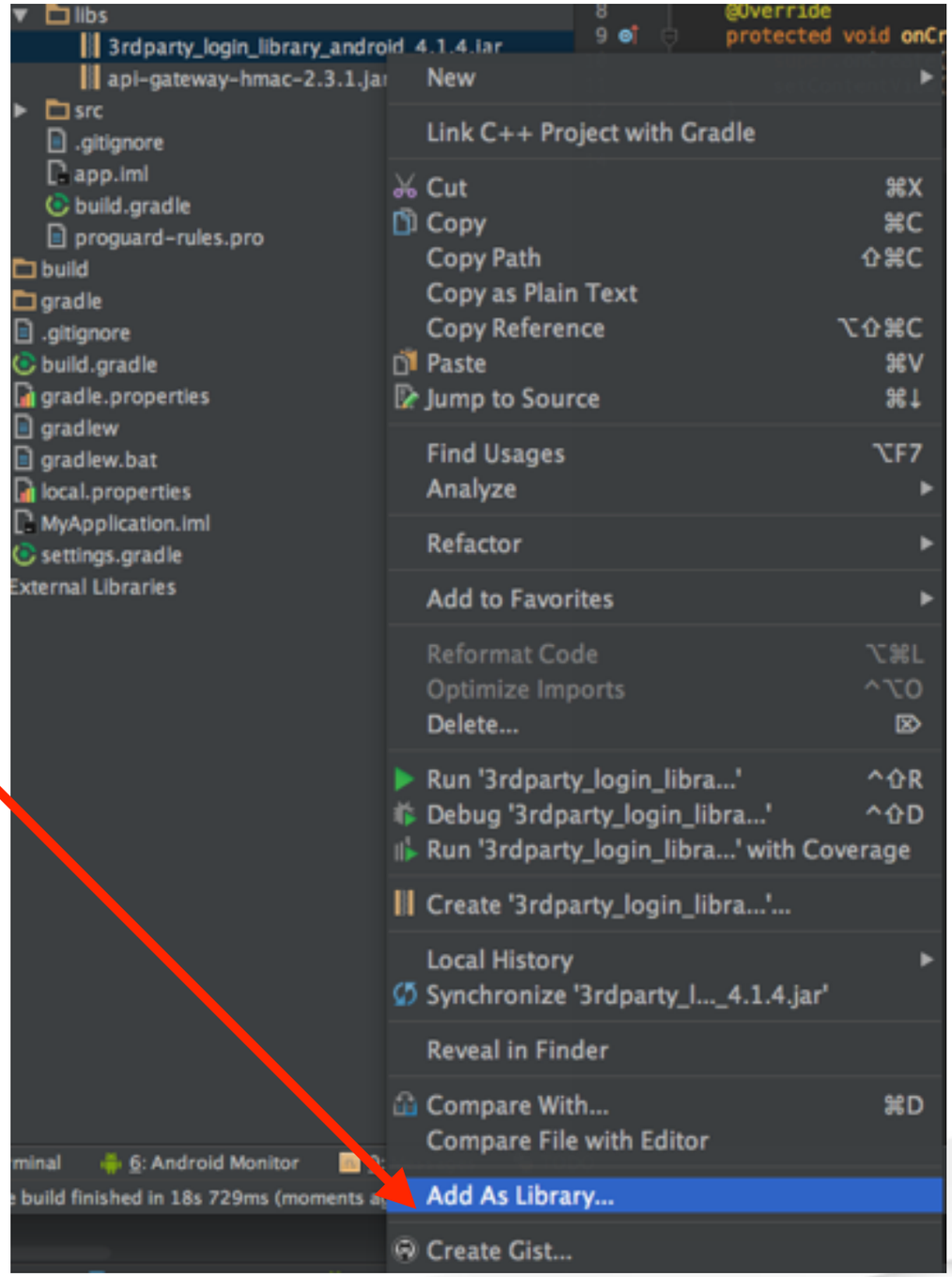
Android Studio libs 설정

아래의 파일들을 아래와 Android Studio에 포함시키며 라이브러리 설정을 한다.

Project 탭을 선택한 후 libs 폴더 내에 아래와 같이 Naver Cafe Plug SDK 파일들을 위치시킨다.
기본적으로 app/libs 폴더 안에 다음의 세개 파일들을 복사하여 넣으면 아래와 같이 보인다.



.jar 파일들을 클릭한 후 오른쪽 마우스 버튼을 누르면 다음과 같이 ‘Add As Library’
라는 문구를 볼 수 있다. 라이브러리 파일로 사용하도록 진행하여 라이브러리로 설정한다.



build.gradle 설정

라이브러리 설정 후에는 아래와 같이 build.gradle 파일을 설정한다.

```
repositories {
    flatDir {
        dirs 'libs'
    }
}

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

    compile(name:'cafeSdk-2.2.1', ext:'aar')
    compile files('libs/api-gateway-hmac-2.3.1.jar')
    compile files('libs/3rdparty_login_library_android_4.1.4.jar')
    compile 'com.github.bumptech.glide:glide:3.6.1'
    compile 'com.squareup:otto:1.3.8'
    compile 'com.navercorp.volleyextensions:volleyer:2.0.1'
}
```

minSdkVersion 설정

Naver Cafe Plug SDK는 최소 SDK 버전을 16으로 설정하여야 한다.
기존에 16 이하로 설정되어 있을 경우 설정값을 16 이상으로 수정한다.

Naver Cafe Plug SDK API

Naver Cafe PLUG SDK 관련된 API 들은 아래와 같다.

초기화 API(국내 카페용)

아래의 함수는 국내 카페와 관련하여 설정을 초기화 하는 함수이다.
아래의 파라미터에서 사용되는 값들은 네이버 개발자 센터(https://developers.naver.com/main/)에 미리 등록되어 있어야 한다.
사업부등에 아래와 같은 값들을 등록하기를 요청한 후 해당 값을 전달받아 입력하도록 한다.

```
CRSDKManager.initNaverCafeSDK("197CymaStozo7X5r2qR5", "evCgKH1kJL", 28290504);
```

Parameters

```
NSString *clientId
: 클라이언트 아이디

NSString *clientSecret
: 클라이언트 Secret 값

NSInteger cafeId
: 카페 아이디
```

초기화 API(글로벌 채널용)

글로벌 채널을 위한 설정을 초기화 하는 함수이다. 글로벌 채널을 미지원 시에는 사용하지 않아도 된다.

```
CRSDKManager.initNaverCafeSDKGlobal("197CymaStozo7X5r2qR5", "evCgKH1kJL", 28290504);
```

Parameters

```
NSString *consumerKey
: 클라이언트 아이디

NSInteger cafeId
: 카페 아이디
```

초기화 API(글로벌 채널용)

화면에 Naver Cafe 위젯(Widget) 아이콘을 표시할지 말지에 대하여 설정하는 함수이다.
아래의 함수에서 true 로 설정하여 호출하면 화면 측면에 위젯 아이콘이 표시되어 카페 커뮤니티 접근 및 게시글, 화면 캡쳐, 영상 녹화 등의 기능을 사용할 수 있다.

```
CRSDKManager.showNaverCafeWidget(true);
```

Parameters

`boolean isShow`
: Widget 을 표시할 경우 true 로 설정

Naver Cafe SDK Support

네이버 카페 제휴 신청 : https://github.com/naver/cafe-sdk-ios/wiki/%5B한%5D-선행-작업
글로벌 채널 제휴 신청 : https://github.com/naver/cafe-sdk-ios/wiki/%5B한%5D%20글로벌%20네이버%20카페%20사용
PLUG 공식 카페 : http://cafe.naver.com/navercafesdk(기본 소개 및 각종 개발 문의)

문의처

SDK 연동 시 문의 사항 및 버그 제보 등은 아래의 이메일로 연락하시기 바랍니다.

당담 부서 : 플랫폼 서비스실
당담자 : 김정훈
이메일 : billkim@curoholdings.com

코드 목록표(CRConfig.h 참조)

onFinishedBilling 이벤트 수신 시 받을 수 있는 code 목록표

Code	Description
0	구매 성공, CRBILLING_SUCCESSED
-1	구매 실패, CRBILLING_FAILED
-2	구매 취소, CRBILLING_CANCELED

onReceivedResponseCode 이벤트 수신 시 받을 수 있는 code 목록표

Code	Description
0	정상 코드값
-10000	이벤트 대상 아님(게스트 또는 대상 아님)
-10100	쿠폰 이벤트가 존재하지 않음
-10200	잘못된 쿠폰 코드
-10201	이미 사용한 쿠폰 코드
-10202	계정 당 쿠폰 사용 횟수 초과
-10301	빈 쿠폰 코드값
-10401	아이템 지급 서버 응답 오류

onFinishedBannerPopup 이벤트 수신 시 받을 수 있는 값 목록표

String	Description
login	CRConstants.BANNER_TYPE_LOGIN, 로그인 배너일 경우
exit	CRConstants.BANNER_TYPE_EXIT, 종료 배너일 경우

코드 목록표(CRConfig.h 참조)

onFailedLogin 이벤트 수신 시 받을 수 있는 code 목록표

Code	Description	References
-1000	네트워크 연결 오류	현재 단말기의 네트워크가 연결되지 않았을 경우
-1001	로그인 요청 실패	플랫폼 서버로의 요청 파라미터 이상 등
-1002	로그인 요청 후 응답값 이상	플랫폼 서버 접속 이상 및 응답 정보 파싱 에러 등
10001	GPGS 재연결 필요 시	해당값은 0.9.7 버전부터 보내지 않음.(Deprecated) 자세한 내용은 아래의 Google GamesActivityResultCodes 값 참고 https://developers.google.com/games/services/android/api/com/google/android/gms/games/GamesActivityResultCodes
10002	GPGS 로그인이 실패한 경우(테스트 상태에서는 테스트 계정 확인 필요)	
10003	GPGS 의 게임이 유저에게 허가되지 않을 경우	
10004	GPGS 설정 이상(주로 keystore 연결 설정 이상)	
10005	GPGS 게임 서비스에서 실시간 게임에서 ‘leave the room’이 선택된 경우	

CRUser 주요 변수

유저의 정보를 담고있는 객체

Name	Type	Description	Case
uid	String	소셜 연동 시의 유저 아이디	Game Center, Google Play, Facebook, Google 등의 UID
provider	String	소셜 제공자 이름	gamecenter, googleplay, guest, facebook, etc.
gameuid	String	플랫폼 서버 유저 아이디	5000001 등의 아이디 번호
name	String	소셜 연동 시의 유저 이름	소셜 플랫폼(Facebook, Google+) 등에서의 유저 이름
market	String	스토어(마켓) 이름	apple, play, one
isAgreedPush	boolean	푸시 수신 동의 설정값	푸신 수신 동의 시 YES 아닐 시 NO
isAgreeNightPush	boolean	야간 푸시 수신 동의 설정값	야간 푸신 수신 동의 시 YES 아닐 시 NO