

# **VOLTUS Client SDK for Unity3d**

플랫폼 서비스실(외부 배포용)

# 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 Unity3d 에서의 셋팅을 위하여 가이드한다.

## 문서 이력

날짜	리비전	설명
2016.05.04	1.0	가이드 문서 최초 생성 및 배포
2016.05.18	1.1	Unity3d Native 형식의 플러그인 형태로 재구성
2016.06.30	1.2	약관, 공지, 배너 팝업 UI 적용
2016.07.13	1.3	0.8.3 버전 가이드 수정, didPurchaseItemLog 리스너 함수 추가
2016.08.22	1.4	0.8.4 버전 가이드 수정
2016.08.25	1.5	0.8.5 버전 가이드 수정
2016.09.05	1.6	코드 목록표 추가
2016.09.12	1.7	0.8.6 버전 가이드 수정, Appendix 최초 생성

# SDK 버전 이력

## 버전 이력

날짜	버전	설명
2016.05.04	0.8.0	최초 베타 버전
2016.05.18	0.8.1	Unity3d Native용 라이브러리 재구성
2016.06.30	0.8.2	약관, 공지, 배너 팝업 UI 적용, 기타 버그 수정
2016.07.13	0.8.3	결제 통계용 로그 API 추가, 기타 버그 수정
2016.08.22	0.8.4	로그인 시 푸시 코드 전달 추가, 무료 충전소 기능 추가, 푸시 알람 버그 수정, 푸시 알람 팝업 UI 변경, 기타 개선 등
2016.08.25	0.8.5	무료 충전소 팝업 출력 API 추가
2016.09.12	0.8.6	푸시 수신 동의 설정 API 추가

# 목차

---

## 1. SDK 소개

1.1 SDK 목적 .....	1
1.2 주요 컴포넌트 .....	2
1.3 각 컴포넌트 세부 사항 .....	3
1.4 SDK 기본 흐름도 .....	6

## 2. Getting Start(시작하기)

2.1 개발 요구 사항 .....	7
--------------------	---

## 3. Quick Guide(빠른 설정 가이드)

3.1 SDK 패키지 파일 импорт .....	8
3.2 CRSettings Prefab 추가 .....	9
3.3 CRListener Prefab 추가 .....	10
3.4 iOS 플랫폼 초기화 .....	11
3.5 Android 플랫폼 초기화 .....	13
3.6 초기 설정 .....	18
3.7 이벤트 델리게이트 목록 .....	20

# 목차

---

## 4. 주요 API 목록

4.1 소셜 인증 .....	21
4.2 약관 페이지 .....	22
4.3 배너 팝업 .....	23
4.4 공지 팝업 .....	24
4.5 유저 프로필 .....	25
4.6 게임 서비스(미지원) .....	26
4.7 결제 로그 .....	27
4.8 무료 충전소 .....	28
4.9 푸시 수신 동의 설정 .....	30

## 5. 마무리

5.1 SDK 관련 문의 .....	31
5.2 문제 해결(Trouble Shooting) .....	32
5.3 코드 목록표 .....	33

## 6. Appendix

6.1 SDK 내 객체 설명 .....	34
-----------------------	----

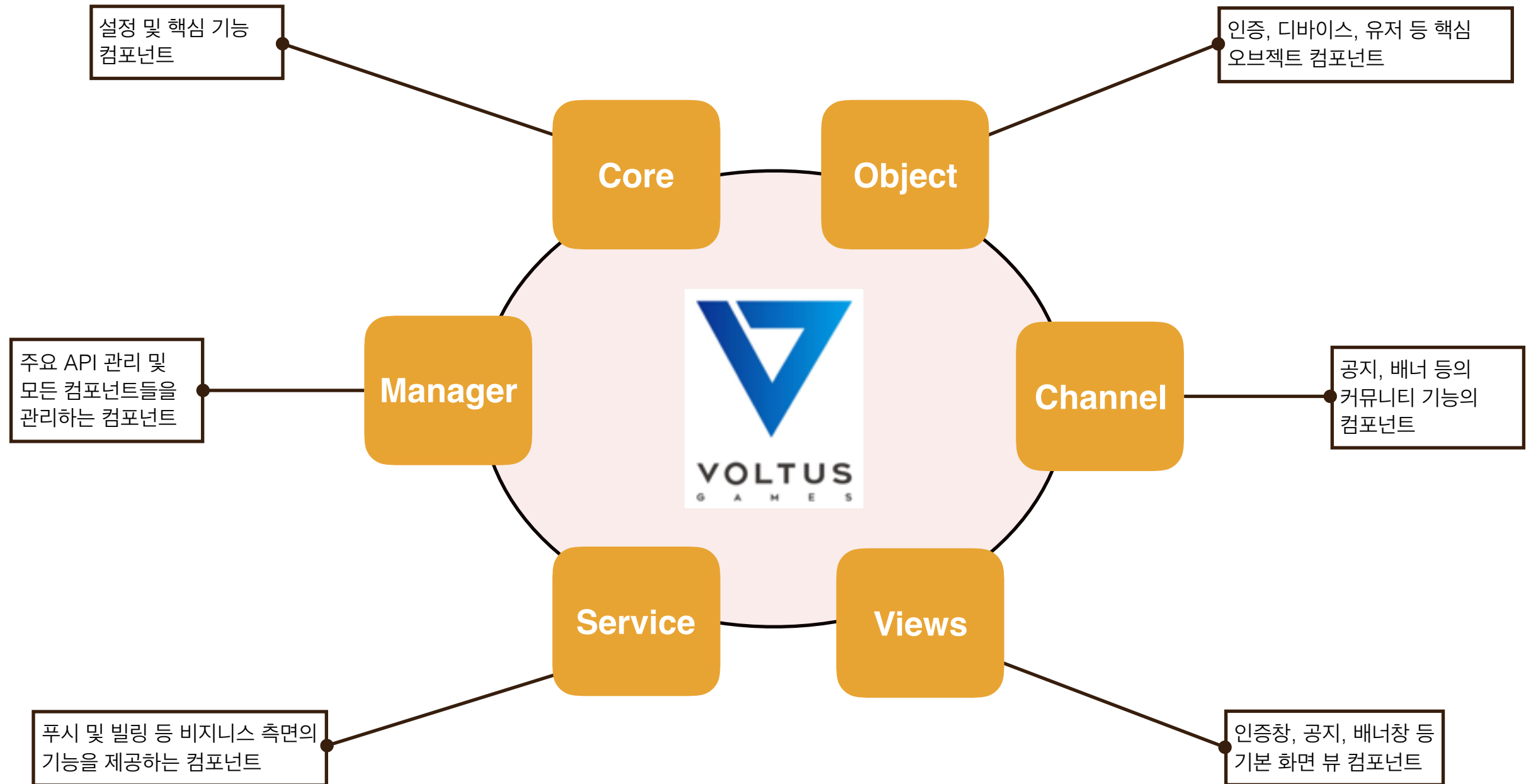
## VOLTUS SDK 목적

VOLTUS SDK는 게임 개발 시 필요한 인증, 공지 등의 사용자 관리 및 운영 측면에서의 기능 들을 제공해주며 게임 개발사는 게임 콘텐츠 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
무료 충전소	VOLTUS 플랫폼의 게임 간의 크로스 프로모션 등을 위하여 자사의 타게임을 추가적으로 설치하면 보상 등을 지급하는 서비스

## 주요 컴포넌트 구성



**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	CRObject	Object Component 들을 위한 기반 클래스
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

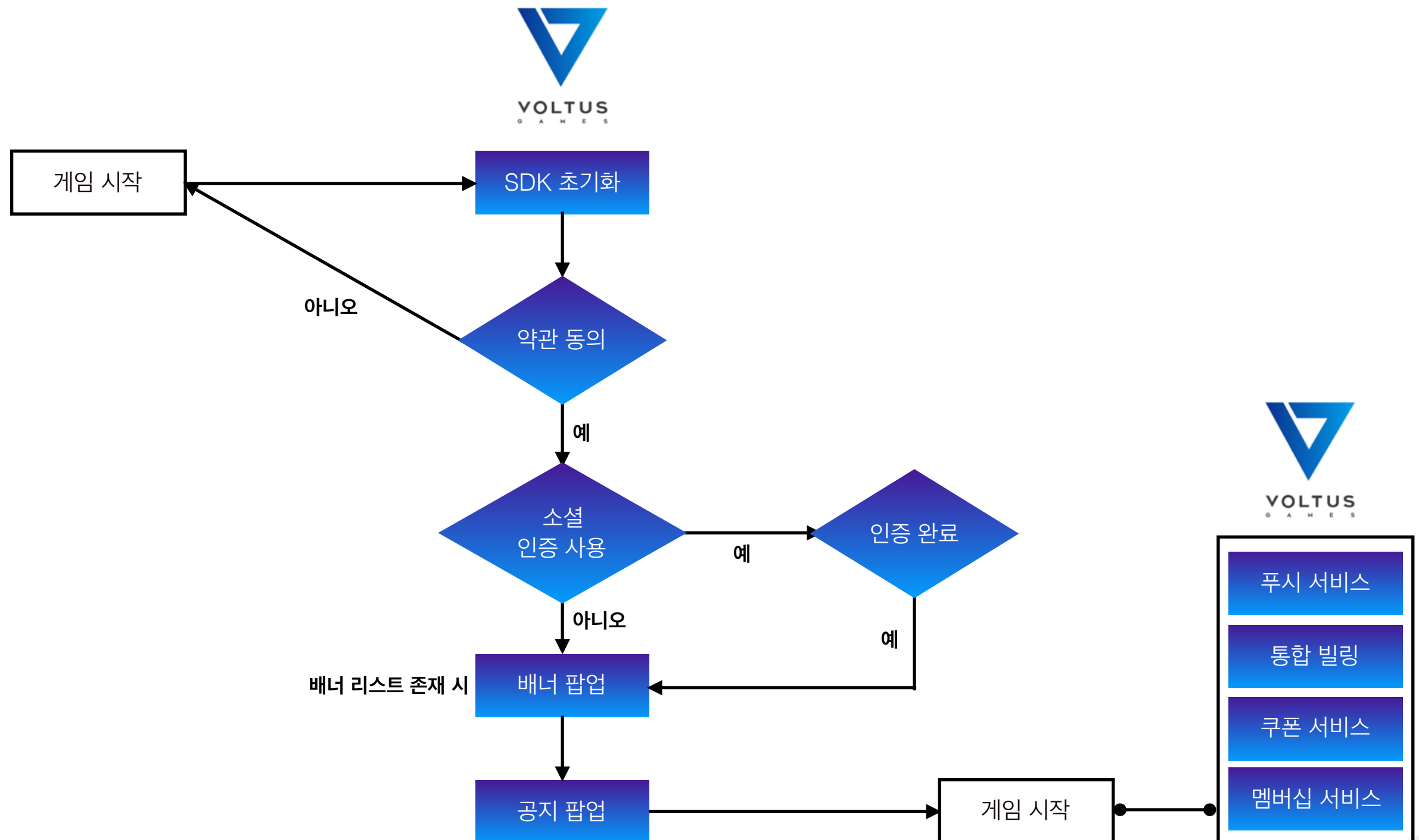
**Component : Service**

Component	Class / File	기능
Service	CRPush	푸시 설정 및 기능을 제공하는 클래스
Service	CRBilling	통합 빌링을 설정하고 기능을 제공하는 클래스(차후 지원 예정)

## Component : Views

Component	Class / File	기능
Views	LoginViewController	소셜 로그인을 선택할 수 있도록 제공하는 뷰 클래스
Views	LoginAuthViewController	소셜 인증을 처리하는 뷰 클래스
Views	TermsViewController	약관을 표시하고 처리하는 뷰 클래스
Views	CRWebViewController	공지를 위한 웹뷰 형식의 뷰 클래스
Views	CRBannerViewController	배너를 표시하기 위한 뷰 클래스

## SDK 전반적인 흐름도



### 기본 개발 요구 사항

Unity3d 4.x later

### 기본 기능 사용 요구 사항

VOLTUS 플랫폼 서버에 등록된 게임 ID  
개발자 관련 스토어 기본 등록 정보들

### 소셜 인증 사용 시

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

Facebook 인증 사용 시 : Facebook 기본 앱 생성(<https://developers.facebook.com/>), 안드로이드 경우는 키스토어의 키해시 등록

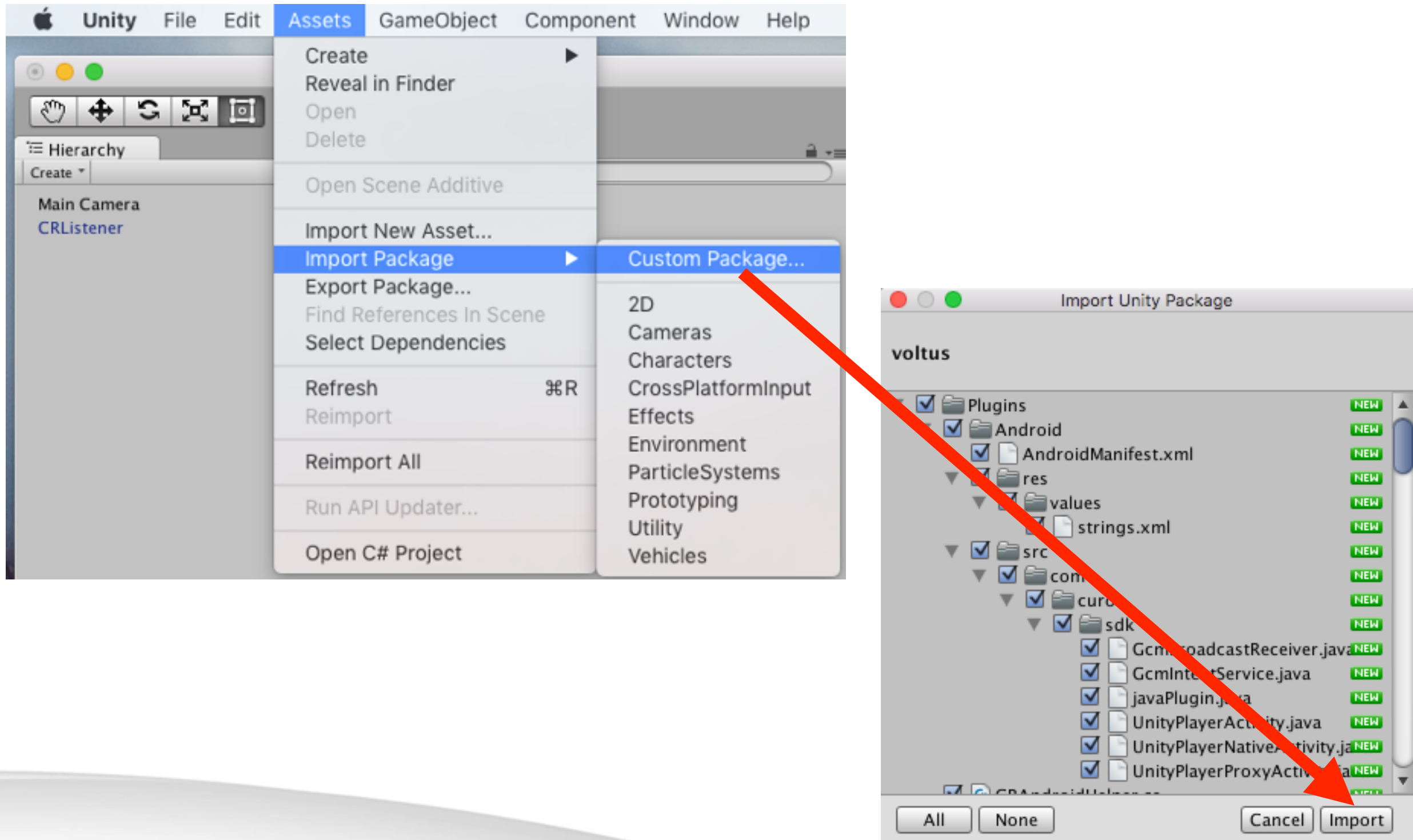
### Unity3d 기본 연동 절차

1. Unity3d 용 SDK 패키지 파일을 임포트한다.
2. CRLListener, CRSettings prefab 파일을 씬에 추가한다.
3. 씬에 추가한 CRSettings 오브젝트에서 게임 아이디, 서비스 모드 등의 설정값을 셋팅한다.
4. 각 플랫폼(iOS, Android)에서 빌드를 한다.

## SDK 패키지 파일 импорт

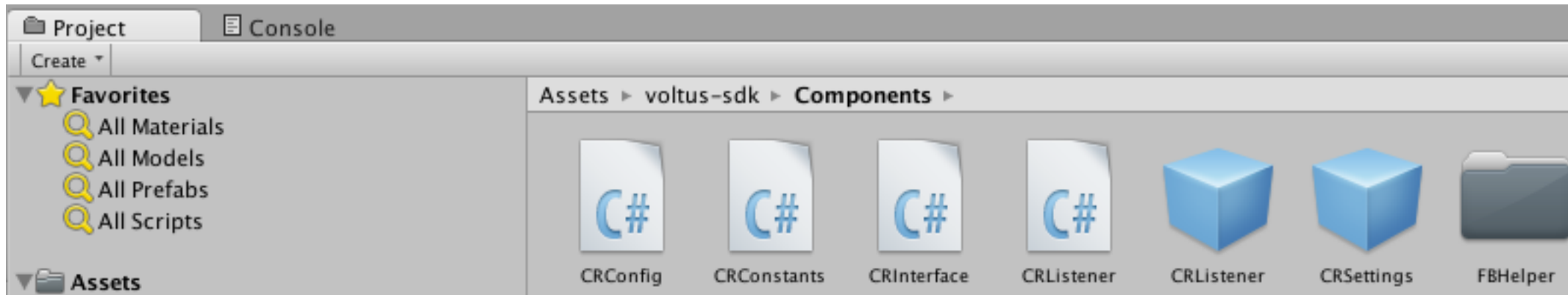
아래와 같은 스텝으로 전달된 SDK 관련 파일 중 SDK 패키지 파일(voltus.unitypackage)을 импорт 시킨다.

**VOLTUS SDK** 안에는 기본적으로 **Facebook SDK** 도 포함되어 있다. 이미 기존에 사용하고 있는 **Facebook SDK** 가 있다면 импорт에서 제외하여도 무방하다.



## CRSettings Prefab 추가

임포트를 진행하면 'Assets/Voltus-SDK/Components' 폴더가 생기며 아래와 같은 파일들이 추가된다.

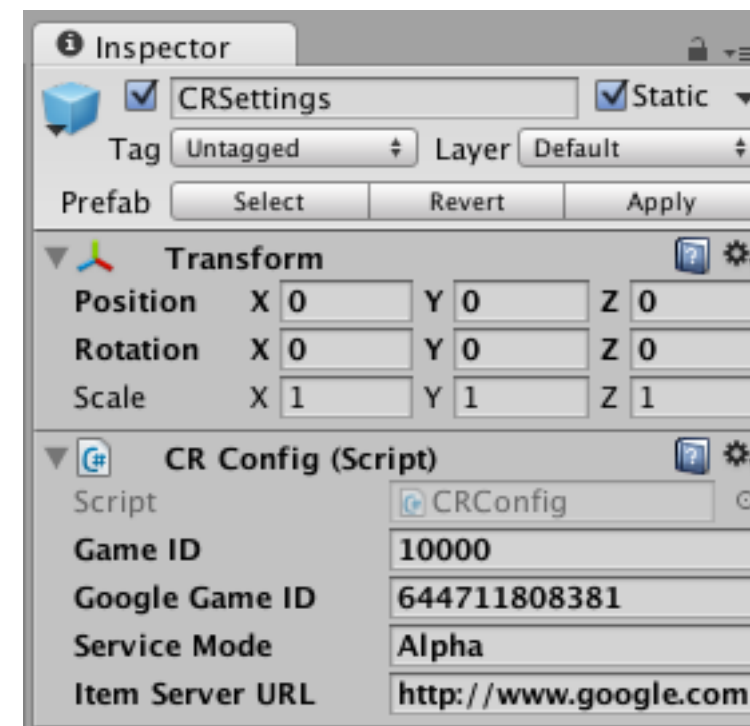
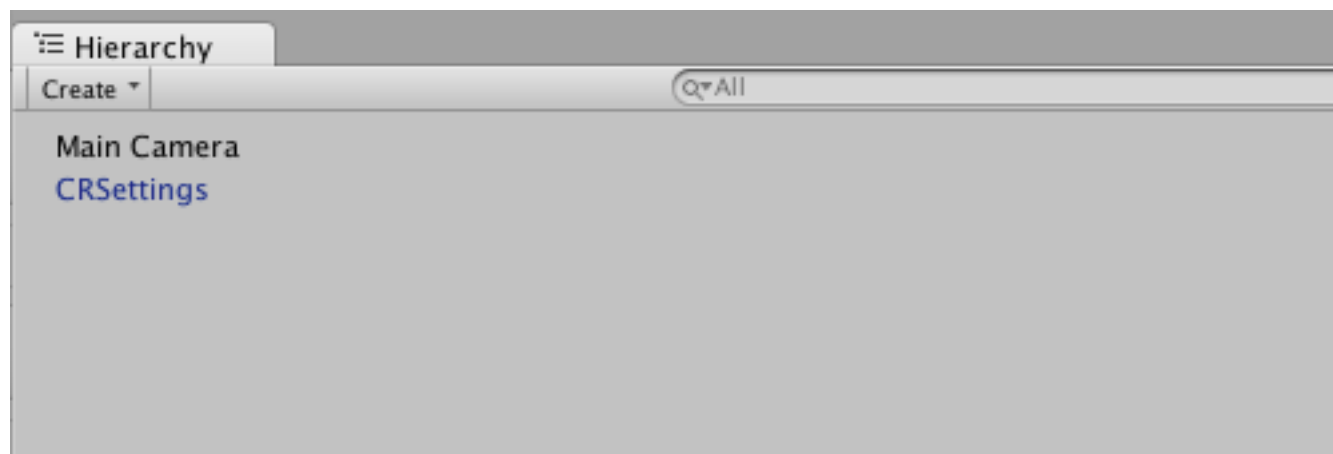


그 중에서 CRSettings.prefab 파일을 씬 오브젝트에 끌어다 놓아 씬에 추가한다.  
추가한 CRSettings 오브젝트를 클릭하여 Inspector에서 게임 아이디, 서비스 모드 등의 설정 값을 설정한다.

**Game ID** : VOLTUS 플랫폼 게임 아이디

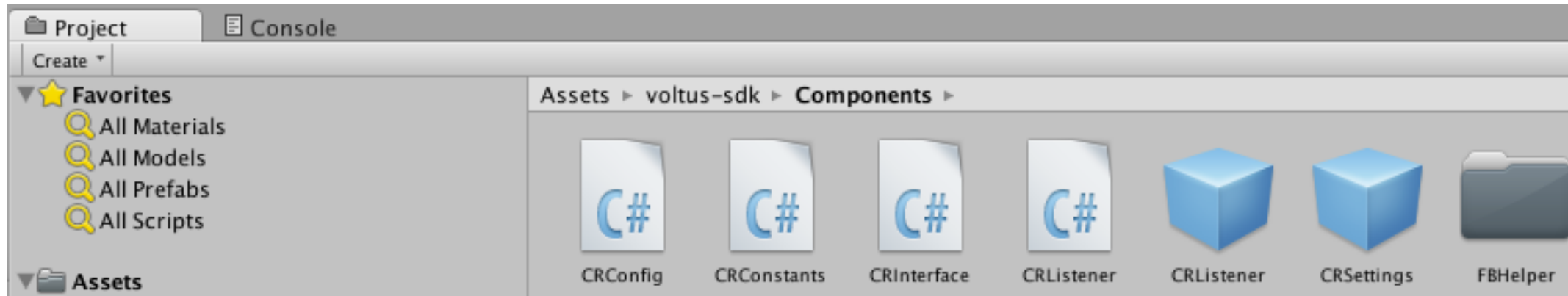
**Google Game ID** : 구글 플레이 게임 아이디(푸시 서비스를 위하여 필요)

**Item Server URL** : 무료 충전소와 관련하여 아이템을 지급하는 서버의 URL 주소

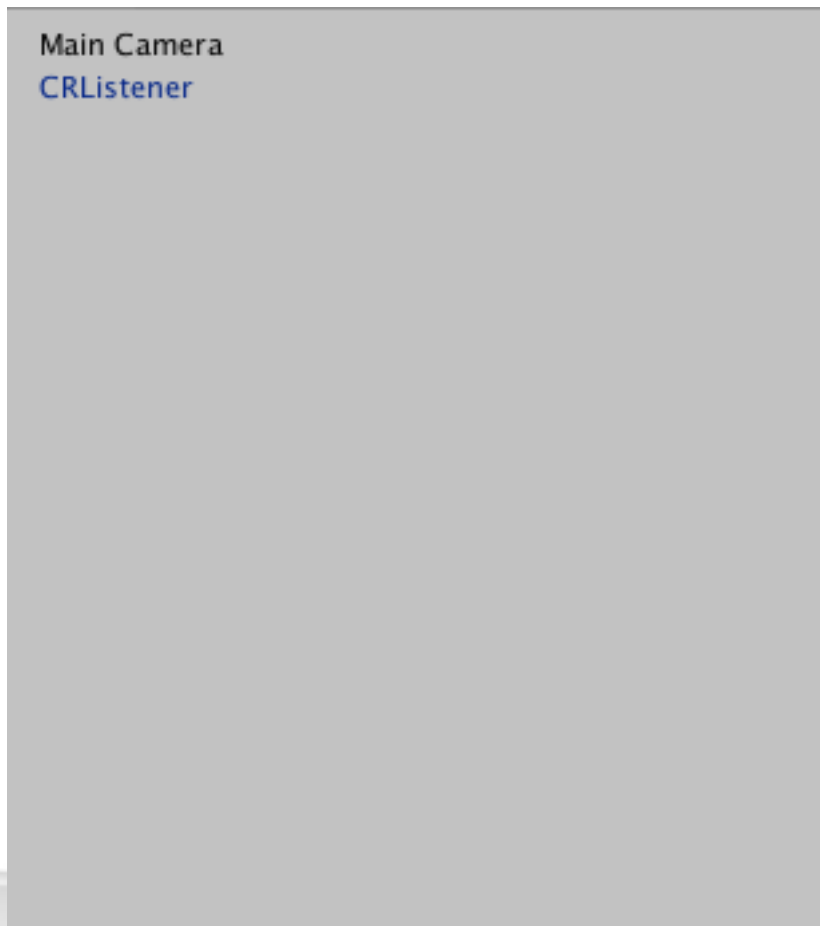


### CRListener Prefab 추가

임포트를 진행하면 'Assets/Voltus-SDK/Components' 폴더가 생기며 아래와 같은 파일들이 추가된다.



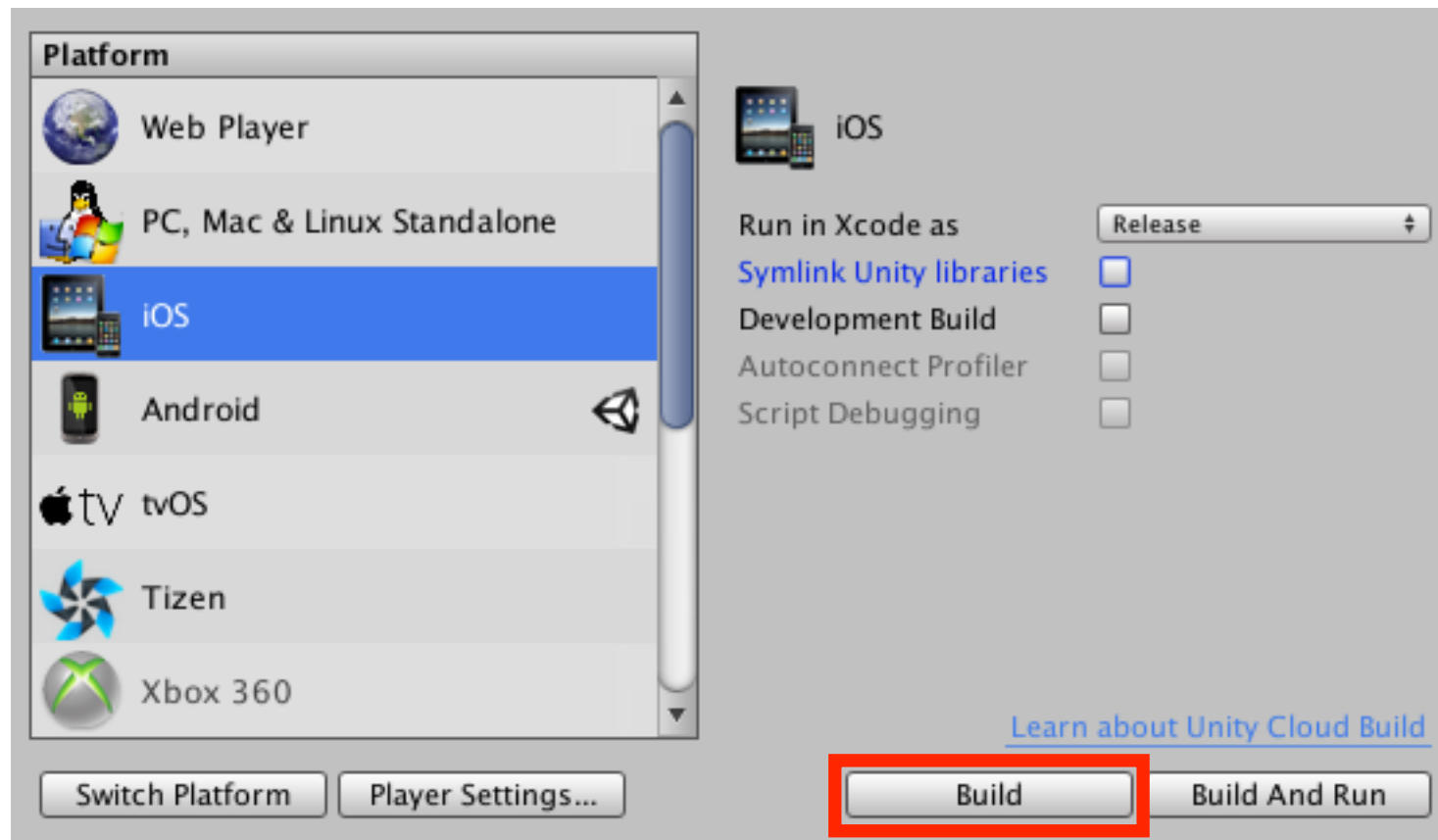
그 중에서 CRListener.prefab 파일을 씬 오브젝트에 끌어다 놓아 씬에 추가한다.



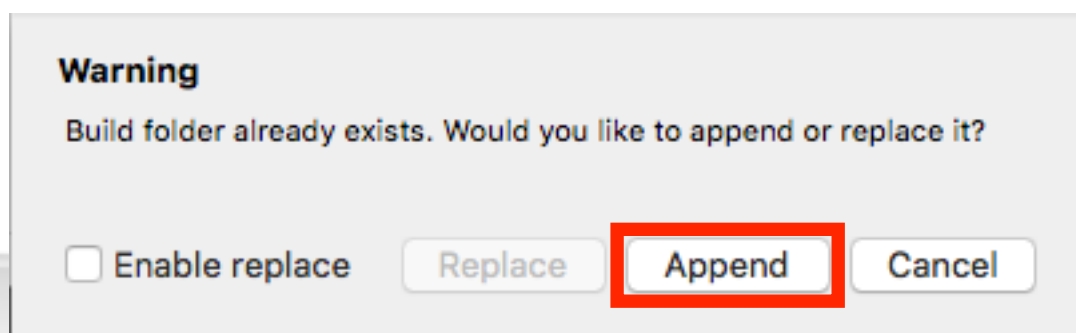
## iOS 플랫폼 설정

## iOS 빌드 및 실행

Unity3d Menu에서 File -> Build Setting을 선택하면 아래와 같은 화면이 나온다. 여기서 Platform을 iOS를 선택하고 Build 또는 Build And Run을 실행하면 Xcode 프로젝트가 생성된다. 처음 실행시에는 다음 페이지에서 진행할 리소스 파일 Xcode 내의 임포트를 위해서 'Build'를 선택하여 먼저 Xcode 프로젝트를 추출한다.



Build And Run을 실행하고 기존 Xcode 프로젝트가 존재하는 경우는 아래와 같은 팝업이 뜬다. 해당 팝업에서 Append를 설정한다.



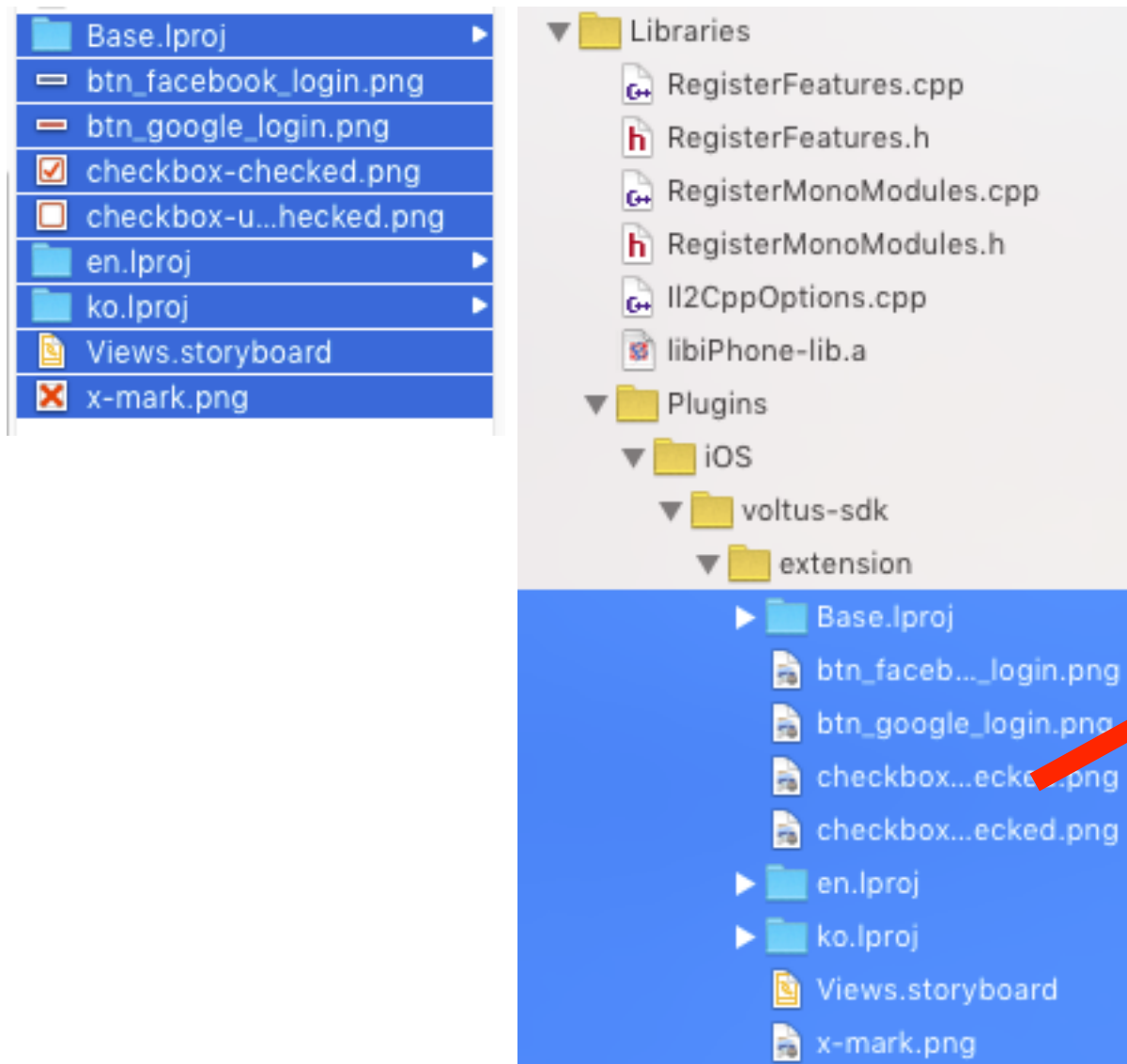


## iOS 플랫폼 설정

## iOS Xcode 프로젝트 내에 일부 리소스 импорт

Xcode 프로젝트가 생성되면 Xcode 프로젝트를 실행하여 아래와 같이 res 폴더에서 **Views.storyboard** 파일을 프로젝트 내에 해당 폴더로 끌어다 놓아 импорт 시킨다.  
해당 리소스 파일(Views.storyboard)은 전달받은 파일의 'Plugins/iOS/voltus-sdk/extension/res' 폴더에 있다.

**중요) 0.8.1 이전에 기존에 아래와 같은 파일들(\*.lproj, \*.png 등)을 импорт 시켰다면 우선 프로젝트에서 삭제하여 준다.**

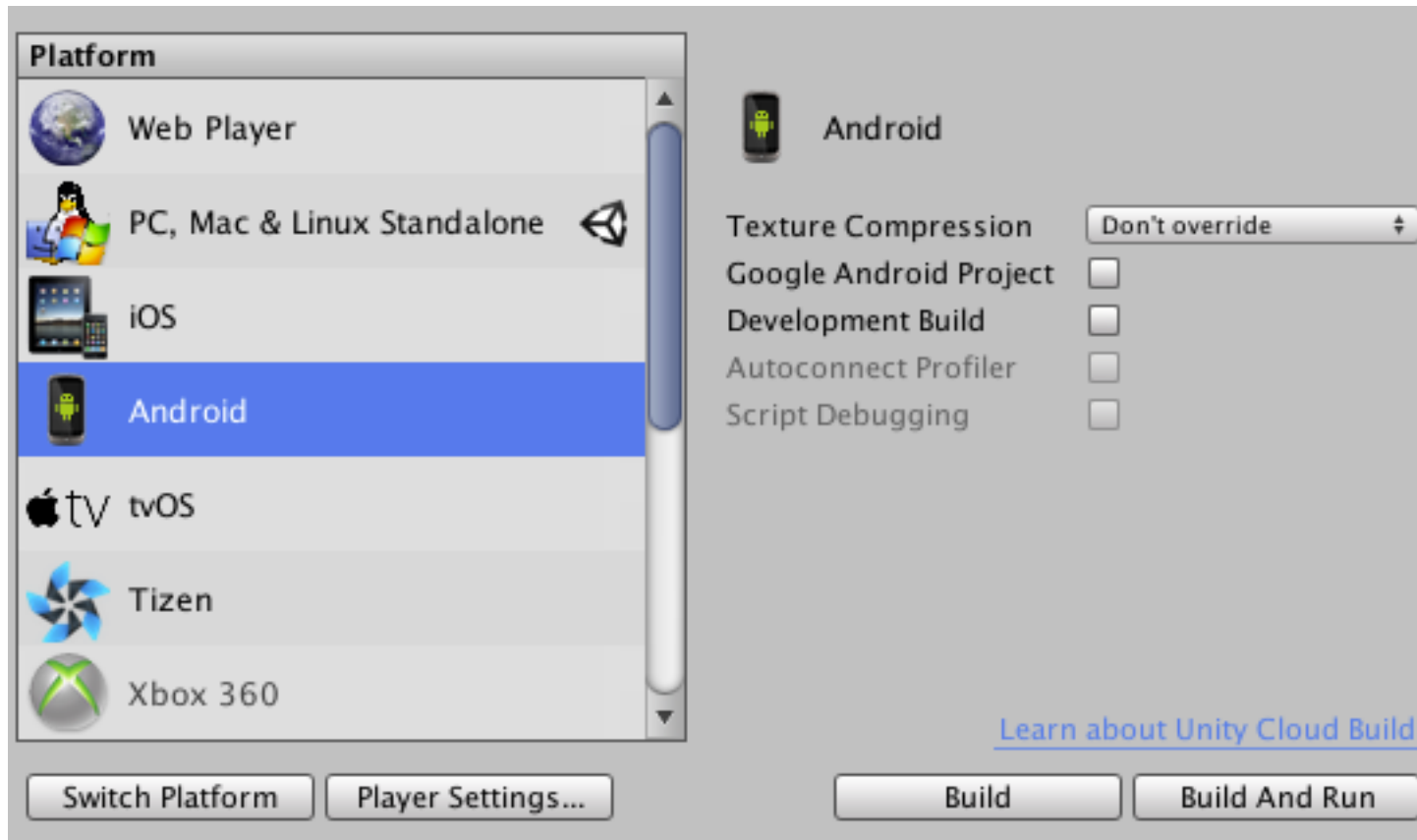


프로젝트에서 импорт 해제

## Android 플랫폼 설정

### Android 빌드 및 실행

Unity3d Menu에서 File -> Build Setting을 선택하면 아래와 같은 화면이 나온다. 여기서 Platform을 Android를 선택하고 Build 또는 Build And Run을 실행하면 APK 파일이 생성된다.



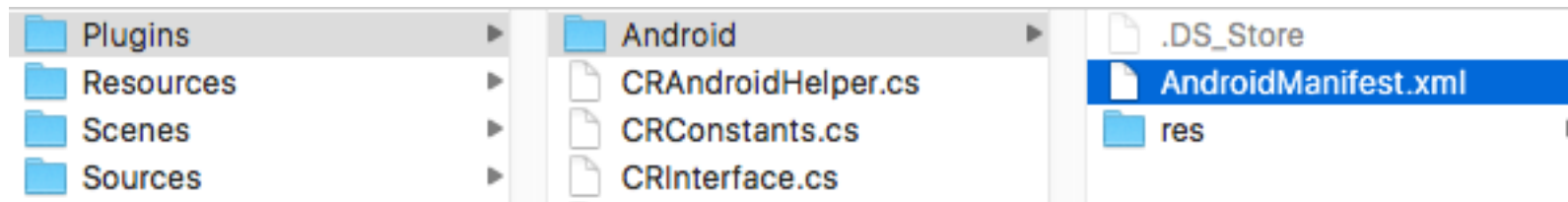
## Android 플랫폼 설정

### AndroidManifest.xml 파일 설정

기존에 사용하던 AndroidManifest.xml 파일을 복사하여 Assets/Plugins/Android 폴더로 붙여넣기한다.

앞으로는 AndroidMainfest.xml 파일을 이 곳에 있는 파일로 관리하며 사용한다.

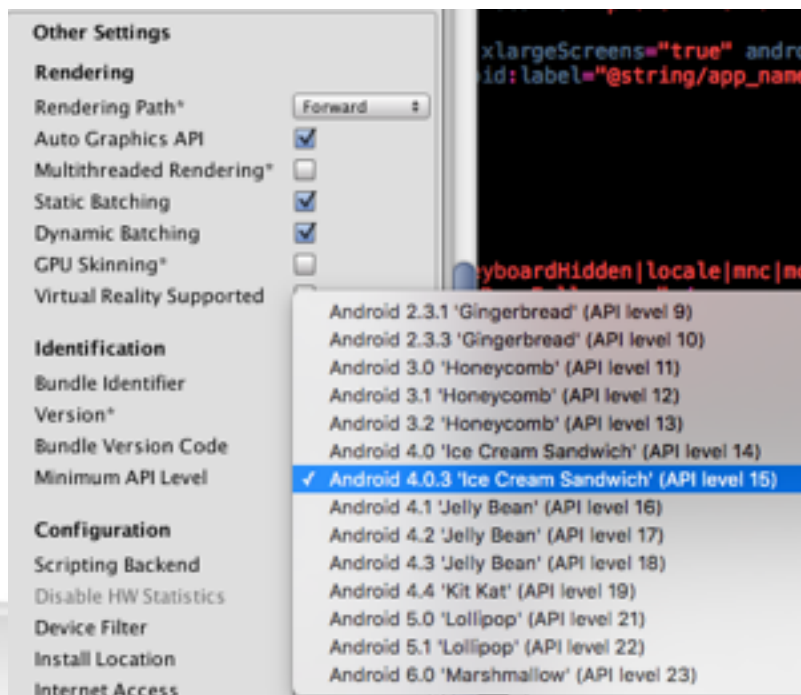
AndroidMainfest.xml 파일 내에서 기본 Activity를 아래와 같이 수정하여 준다.(푸시 서비스를 위하여, [com.curo.sdk.CRUnityPlayerActivity](#))



```
<activity android:name="com.curo.sdk.CRUnityPlayerActivity" android:label="@string/app_name">
```

### Minimum API Level 설정

Android Build를 실행하기 전에 Player Settings 에서 Other Settings 부분에서 Minimum API Level을 15 이상으로 설정한다.



## AndroidManifest.xml 파일 설정

## 1) Permission 설정

## 필수 설정

Assets/Plugins/Android/AndroidManifest.xml 내에 기본 Permission 설정

```
<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"/>
```

## AndroidManifest.xml 파일 설정

## 2) Layout Activity 설정

Assets/Plugins/Android/AndroidManifest.xml 내에 기본 Layout Activity 설정

포함해야 하는 Activity : CRLoginActivity, CRWebviewActivity, CRBannerActivity, PushPopupActivity, TermsActivity

```
<application
.....

    <activity
        android:name="com.curo.sdk.layout.LoginActivity"
        android:screenOrientation="landscape"
        android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
        android:configChanges="keyboardHidden|orientation|screenSize" >
    </activity>

    <activity
        android:name="com.curo.sdk.layout.CRWebviewActivity"
        android:screenOrientation="landscape"
        android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
        android:configChanges="keyboardHidden|orientation|screenSize" >
    </activity>

    <activity
        android:name="com.curo.sdk.layout.CRBannerActivity"
        android:screenOrientation="landscape"
        android:theme="@android:style/Theme.Translucent"
        android:configChanges="keyboardHidden|orientation|screenSize" >
    </activity>

    <activity
        android:name="com.curo.sdk.layout.PushPopupActivity"
        android:screenOrientation="landscape"
        android:configChanges="keyboardHidden|orientation|screenSize" >
    </activity>

    <activity
        android:name="com.curo.sdk.layout.TermsActivity"
        android:screenOrientation="landscape"
        android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
        android:configChanges="keyboardHidden|orientation|screenSize" >
    </activity>
.....
</application>
```

## AndroidManifest.xml 파일 설정

## 3) 푸시 서비스 설정(옵션)

AndroidManifest.xml 파일 내에서 기본 Activity를 아래와 같이 수정하여 준다.(푸시 서비스를 위하여, `com.curo.sdk.CRUnityPlayerActivity`)

```
<activity android:name="com.curo.sdk.CRUnityPlayerActivity" android:label="@string/app_name">
```

푸시 서비스 이용 시에는 아래의 Permission과 Receiver, Service 등을 추가한다.

```
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.GET_TASKS"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<permission android:name="패키지명.permission.C2D_MESSAGE"
    android:protectionLevel="signature" />
<uses-permission android:name="패키지명.permission.C2D_MESSAGE" />
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
```

```
<application
.....
<receiver android:name="com.curo.sdk.GcmBroadcastReceiver"
    android:permission="com.google.android.c2dm.permission.SEND">
    <intent-filter>
        <!-- Receives the actual messages. -->
        <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
        <!-- Receives the registration id. -->
        <action android:name="com.google.android.c2dm.intent.REGISTRATION"/>
        <category android:name="패키지명"/>
    </intent-filter>
</receiver>

<service android:name="com.curo.sdk.GcmIntentService" />
.....
</application>
```

### 플랫폼 초기화 선언

원하는 씬에서 아래의 초기화 코드를 Start() 등의 함수 등에서 선언한다.

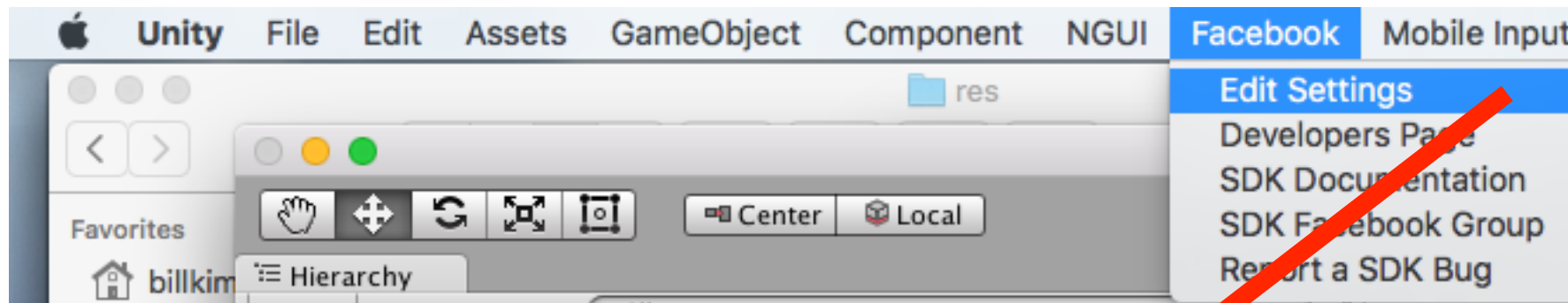
```
using VOLTUS.SDK;

public class SampleScene : MonoBehaviour
{
    void Start ()
    {
        CRSDKManager.initSDK(true);
    }
}
```

### Facebook 초기 설정(옵션)

Facebook 소셜 인증을 위해서는 아래와 같은 설정이 필요하다.

Unity 상단 메뉴에서 Facebook -> Edit Settings 를 선택하여 Inspector에서 각종 이름 및 앱 아이디 등의 설정값을 제대로 입력한다.





## 이벤트 델리게이트 목록

CRLlistener.cs 파일 안에서는 아래의 이벤트 델리게이트 함수들이 포함되어 있다.

각 이벤트는 각 상황에 따라서 자동적으로 호출되므로 필요 시에는 해당 이벤트 안에서 원하는 기능을 수행하면 된다.

델리게이트 메소드	설명
didFinishedLogin	로그인 성공 시에 호출되는 델리게이트 함수
didFailedLogin	로그인 실패 시에 호출되는 델리게이트 함수
didAgreedTerms	약관을 모두 동의할 시에 호출되는 델리게이트 함수
didFinishedLogout	로그아웃 성공 시에 호출되는 델리게이트 함수
didFailedLogout	로그아웃 실패 시에 호출되는 델리게이트 함수
didReceiveLinkType	배너 팝업에서 메인 이미지를 터치 시에 호출되는 델리게이트 함수
didReceivePushUrl	푸시 수신 시에 app 타입의 액션이 수신 시 호출되는 함수, web 타입은 해당 함수가 호출되지 않는다.
didPurchaseItemLog	결제가 성공하고 통계를 위하여 로그 API을 송신 후 호출되는 함수, result값이 true이면 로그 전송 성공
didFinishedBannerPopup	모든 배너 출력이 완료되면 호출되는 델리게이트 함수(5.3 코드 목록표 참조)

## 소셜 인증 기능

VOLTUS 플랫폼은 각종 소셜 인증 기능을 지원한다. 버전 별 소셜 인증 기능 지원 목록은 아래와 같다.

버전	지원 소셜 인증
0.8.2	GPGS(Google Play Game Service), Guest(iOS 만)
0.8.1	Facebook, Guest
0.8.0	Facebook, Google+, Guest

## 로그인 API(플랫폼 로그인)

현재 버전(0.8.2)에서는 소셜 인증은 제거되었으며, 개별적인 소셜 인증(예 : GPGS 등) 후에 아래의 함수를 호출하여 플랫폼 로그인을 시도한다.

**해당 로그인 API를 소셜 인증이 성공한 후에 반드시 호출하여야만 정상적인 푸시 등의 기능 이용이 가능하다.**

**pushToken 값은 iOS 만을 위하여 필요하므로 Android는 불필요하다. Android 플랫폼에서는 그냥 입력값을 넣지 않고 "" 등으로 처리해도 된다.**

```
CRSDKManager.login((int) CRConstants.CRPlatformType.CRPlatformTypeGoogle, "uid", "name", "email", "pushToken");
```

## Parameters

**type(필수)**

플랫폼 타입(현재는 GPGS만 가능)

**uid(필수)**

GPGS User ID, 플랫폼 서버에서 유저 식별용

**name(옵션)**

유저 이름, 이름값이 없으면 그냥 비워도 된다.

**email(옵션)**

유저 이메일, 이메일이 없으면 그냥 비워도 된다.

**pushToken(필수)**

푸시 토큰 값

## 로그아웃 API

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

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

### 약관 팝업 출력 API

해당 함수 호출 시 기본 플랫폼 약관 팝업을 출력한다.

해당 함수는 보통 게임 시작과 동시에 한번 호출하는 것을 권장하며 SDK 자체적으로 한번 동의하고 나면 해당 함수를 호출하여도 다시 팝업이 뜨지 않는다.(캐시 저장)

**정확한 호출 시점은 사업부와 협의하여 결정한다.**

```
CRSDKManager.showTermsPopup();
```

### 배너 팝업 출력 API

해당 함수 호출 시 기본 플랫폼 배너 팝업을 출력한다.

반드시 로그인, 게임 종료 팝업은 특정 호출 시점(로그인 후, 게임 종료 등등)에 맞게 함수 호출이 되어야 하므로 반드시 사업부와의 협의를 거친 후 누락되거나 잘못 호출되는 사항없이 적용되어야 한다.

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

#### Parameters

string

BANNER\_TYPE\_LOGIN : 로그인 팝업  
BANNER\_TYPE\_EXIT : 게임 종료 팝업

### 공지 팝업 출력 API

해당 함수 호출 시 기본 플랫폼 공지 팝업을 출력한다.

**해당 함수 호출 시점은 사업부와 협의하여 진행한다.** 보통은 로그인 성공 후 호출하는 것을 권장한다.

현재 SDK는 배너 팝업이 뜨면 모든 배너가 다 뜨고 나서는 공지 팝업이 최종적으로 뜨도록 설정되어 있다.

```
CRSDKManager.showNoticeListPopup(true);
```

#### Parameters

bool

true : 강제로 팝업 출력

false : 하루 보지 않기를 선택했다면 미출력

### 유저 프로필 API

해당 함수 호출 시 유저의 기본 정보를 가져온다.

플랫폼 서버에서의 유저 식별은 CRUser 내의 gameuid로 식별한다. 차후 gameuid 값을 CS 등에서의 처리를 위해 사용하면 된다.

```
CRSDKManager.getUser();
```

#### Return

CRUser

유저의 기본 정보 등의 값이 반환된다.

### 게임 센터 및 구글 플레이 게임 서비스

해당 기능은 현재(0.8.1)에서는 정상 지원되지 않는다.

개발사에서 별도로 게임 센터 및 구글 플레이 서비스를 연동하여야 한다.

### 결제 성공 로그 API

결제가 성공하고 결제 성공에 대한 통계 수집을 목적으로 해당 함수를 호출한다.

```
CRSDKManager.purchaseItemLog("productID", "itemName", "KRW", "1000");
```

#### Parameters

string

상품 아이디(productID)

string

아이템 이름(itemName)

string

통화 코드(ISO) (KRW, JPY 등)

string

상품 가격



### 무료 충전소 시스템

무료 충전소는 기본적으로 공지 사항 팝업 안에 포함되어 있다.

무료 충전소는 현재 게임의 '무료 충전소' 페이지에서 VOLTUS 플랫폼 내의 타게임을 설치하면 현재 게임에서 로그인 시에 체크하여 보상을 지급하는 시스템이다.

기본적으로 필요한 API 는 두가지 이다. 아래의 두 개의 API 반드시 플랫폼 로그인이 성공한 이후 호출하여야 한다.

1. 무료 충전소 조건을 충족하여 보상 지급이 가능한 아이템 리스트(SDK 내부에서 **CRSDKManager.login()** 함수를 호출하면 자동적으로 호출하도록 되어 있다.)
2. 보상 지급을 완료하는 함수

### 보상 아이템 리스트 목록 API(SDK 내부에서 로그인 성공 시에 호출하므로 **호출 불필요**)

```
CRSDKManager.requestRewardList();
```

해당 함수를 호출하면 **CRListenter.cs**에서 **didReceiveResponse** 리스너 함수가 호출되며 해당 리스너 함수 내에서 **CRSDKManager.freeChargeRewardList** 리스트(List)를 구성하게 된다.

차후 로비 등의 화면에서 **보상 아이템 지급 완료 API**를 각각 리스트 아이템 별로 호출하여 지급 완료를 요청하여야 차후 재로그인 시에 이미 지급한 보상 아이템 목록은 오지 않게 된다.

해당 함수는 SDK 내부에서 **CRSDKManager.login()** 함수를 호출하면 자동적으로 호출하도록 되어 있기 때문에 개발사에서 호출할 필요는 없다.

해당 함수 호출 성공 시에는 **didReceiveResponse** 리스너 함수가 자동적으로 호출되며 해당 리스너 함수에서 보상 아이템에 대한 리스트 변수인 **CRSDKManager.freeChargeRewardList** 객체를 구성한다. 해당 리스트 객체 변수의 갯수가 **1개 이상**이 되면 '보상 관련 팝업'을 띄우고 해당 아이템을 보상 지급하면 된다.

보상 확인 팝업에서 '확인' 등의 버튼을 누를 시에 아래의 '보상 아이템 지급 완료 API' 함수를 호출하면 된다.

각 아이템(**CRConfig.CRFreeChargeItem**) 별로 **freeChargeID** 가 있으며 '보상 아이템 지급 완료 API' 호출 시 파라미터로 입력하면 된다.

또한 보상 아이템 팝업의 내용을 위해 **itemCode**, **itemCount** 등의 변수 값을 확인하여 정확히 맞도록 팝업을 띄어 유저에게 보상 아이템의 내용을 알려준다.

**CRSDKManager.freeChargeRewardList** 리스트를 다루는 코드는 **didReceiveResponse** 리스너 함수의 하단부를 참고하여 사용하면 된다.

### 보상 아이템 지급 완료 API

보상을 받았으면 더이상 보상 아이템 리스트를 플랫폼 서버에서 받지 않아야 하므로 **보상 아이템 지급 완료 API**를 호출하도록 한다.  
앞에서 설명한 ‘보상 아이템 리스트 목록 API’ 호출 이후 **CRSDKManager.freeChargeRewardList** 리스트 객체의 크기가 1 이상일 경우는 로비 등의 화면에서 ‘보상 지급 팝업’과 출력과 동시에 호출하도록 한다. 각 리스트 내의 아이템 별로 각각 개별로 호출하여야 한다.

보상 아이템에 대한 타입과 갯수는 CRFreeChargeItem 객체 내의 **itemCode**, **itemCount** 의 코드와 갯수를 사업부와 협의하여 아이템 보상 지급 팝업 노출 시에 정확한 내용을 출력할 수 있도록 한다.

ex) **CRSDKManager.freeChargeRewardList** 리스트에서 개별 아이템 가져오는 예제

```
CRFreeChargeItem newItem = CRSDKManager.getRewardItemAtIndex(0); // 첫 번째 보상 아이템 리스트를 가져오는 코드
```

보상 아이템 지급 완료 API가 정상적으로 호출되면 **didFinishedFreeChargeReward** 리스너 함수가 호출된다.

```
CRSDKManager.requestRewardComplete("freeChargeID");
```

#### Parameters

```
string freeChargeID
```

: 무료 충전 보상 아이템 아이디

### 무료 충전소 팝업 출력 API

아래의 함수를 호출하여 무료 충전소 화면을 출력할 수 있다.

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

### 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 아래와 같이 함수 반환값이 참이면 수정이 성공한 경우이다.

```
if(CRSDKManager.modifyPushPermission(true))
{
    print("modifyPushPermission is succeeded.");
}
```

#### Parameters

bool allow

: 푸시 수신 동의 여부. 수신 동의 시 true

### 문의처

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

담당 부서 : 플랫폼 서비스실

담당자 : 김정훈

이메일 : [billkim@curoholdings.com](mailto:billkim@curoholdings.com)

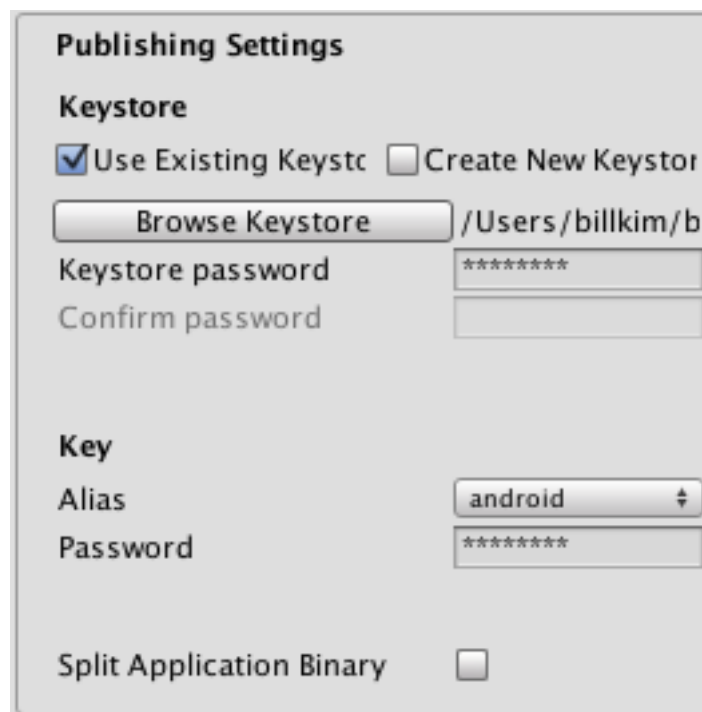
### 이벤트 델리게이트 수신이 안될 때

각종 이벤트 델리게이트들이 수신이 안 될 때는 CRListener 오브젝트를 씬 내에 추가했는지 살펴본다.('3.3 CRListener Prefab 추가' 참고)

### Eclipse Project Export 시에 인증서 에러

Eclipse 용 프로젝트를 Export 하였을 경우 아래와 같은 에러 발생 시에는 키스토어가 제대로 설정되어 있는지 확인한다.  
Android Player Settings에서 아래의 Publishing Settings 와 같은 Keystore 로 Eclipse가 바라보고 있는지 확인한다.

```
[2016-05-18 16:02:01 - UnityPlayerActivity] Installation error: INSTALL_PARSE_FAILED_NO_CERTIFICATES  
[2016-05-18 16:02:01 - UnityPlayerActivity] Please check logcat output for more details.  
[2016-05-18 16:02:01 - UnityPlayerActivity] Launch canceled!
```



## 코드 목록표(CRListener.cs 참조)

**didFinishedBannerPopup** 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

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
gameuid	string	플랫폼 서버 유저 아이디	5000001 등의 아이디 번호
name	string	소셜 연동 시의 유저 이름	소셜 플랫폼(Facebook, Google+) 등에서의 유저 이름
isAgreedPush	bool	푸시 수신 동의 설정값	푸신 수신 동의 시 YES 아닐 시 NO
isAgreedNightPush	bool	야간 푸시 수신 동의 설정값	야간 푸신 수신 동의 시 YES 아닐 시 NO