

VOLTUS Client SDK for iOS

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 iOS 플랫폼을 위하여 가이드한다.

문서 이력

날짜	리비전	설명
2016.05.04	1.0	가이드 문서 최초 생성 및 배포
2016.06.03	1.1	IAP 기능 , 로고 출력, 쿠폰 등록 기능 가이드 추가
2016.06.07	1.2	코드 목록표 추가
2016.06.30	1.3	0.9.1 버전 관련 가이드 업데이트
2016.07.13	1.4	0.9.2 버전 관련 가이드 업데이트
2016.08.08	1.5	0.9.3 버전 관련 가이드 업데이트
2016.08.18	1.6	0.9.4 버전 관련 가이드 업데이트
2016.09.01	1.7	0.9.7 버전 관련 가이드 업데이트, onReceivedResponseCode 코드값 수정, 4.10 무료 충전소 수정

문서 개요

문서 이력

[illegible]

SDK 버전 이력

버전 이력

날짜	버전	설명
2016.05.04	0.8.0	최초 베타 버전
2016.06.03	0.9.0	IAP 기능 추가(애플 앱스토어, 구글 플레이), 로고 출력 기능 추가, 쿠폰 기능 추가
2016.06.30	0.9.1	로고, 약관, 공지, 배너, 로그인 화면 등 UI 개선, 구글 플러스 로그인 추가
2016.07.13	0.9.2	로고 및 약관 UI 개선, 기타 버그 수정
2016.08.08	0.9.3	게임 센터 기본 인증 지원, 무료 충전소 추가
2016.08.18	0.9.4	무료 충전소 수정, 로그인 실패 시 코드 추가
2016.09.05	0.9.7	푸시 수신 동의 설정 API 추가, 공지 사항 On/Off 기능 추가, 무료 충전소 팝업 출력 API 추가 등
2016.09.13	0.9.8	푸시 수신 동의 여부 체크 기능 추가, Facebook Ads 기능 추가, 기타 버그 수정
2016.10.10	0.9.9	야간 푸시 동의 기능 추가, 플랫폼 서버 관련 주소 변경, External(Public) IP 정보를 플랫폼 서버로 전송 추가
2017.02.24	1.0.0	Naver Cafe Plug SDK 연동 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 관련 파일	8
3.2 Xcode 설정	9
3.3 초기화 코드 적용	12
3.4 앱 델리게이트 연결	13
3.5 이벤트 코드 추가	14
3.6 이벤트 델리게이트 목록	15

목차

4. 주요 지원 기능

4.1 소셜 인증	16
4.2 약관 페이지	18
4.3 배너 팝업	19
4.4 공지 팝업	20
4.5 푸시 알림	21
4.6 게임 센터	22
4.7 IAP(인앱)	23
4.8 플랫폼 로고 출력	24
4.9 쿠폰 등록	25
4.10 무료 충전소	26
4.11 푸시 수신 동의 설정	27
4.12 Naver Cafe Plug	28

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	35
-----------------------	----

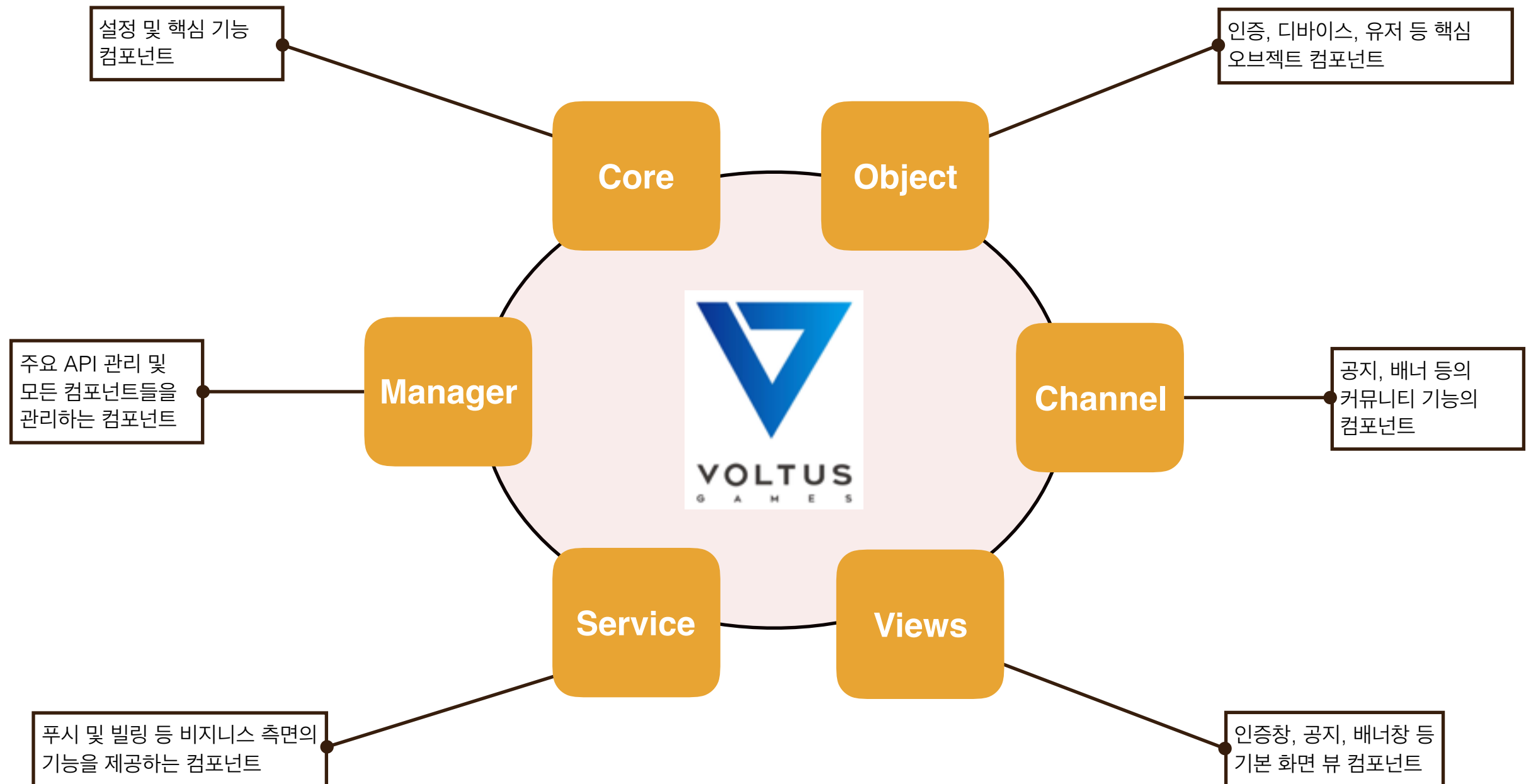
VOLTUS SDK 목적

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
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



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

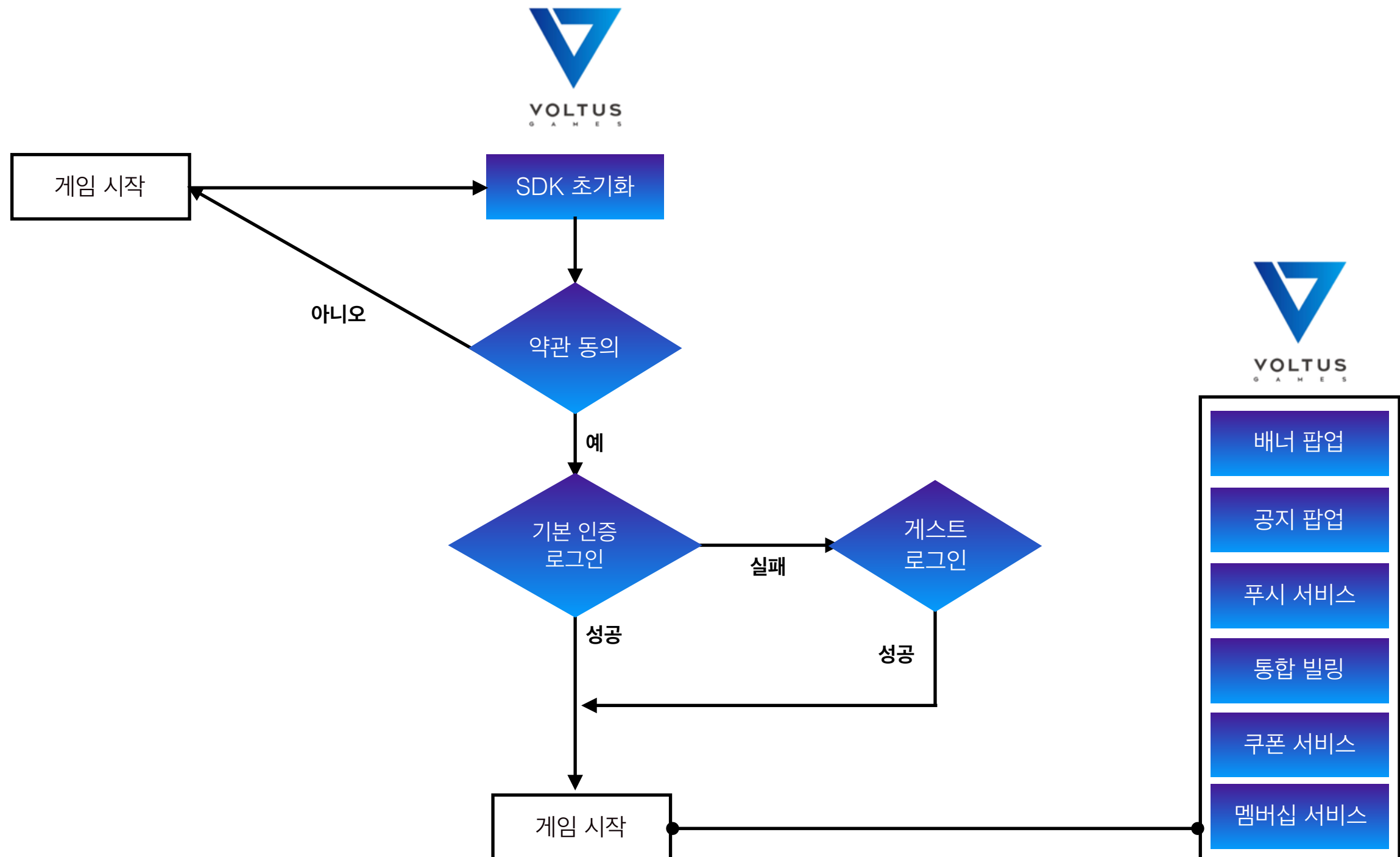
Component : Service

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

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	배너 팝업을 위한 뷰 클래스

SDK 전반적인 흐름도



기본 개발 요구 사항

OS X, Xcode 7.x later, iOS 7.0 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

SDK 관련 파일

SDK 관련 필요 파일은 아래와 같다.

1. CuroSDK.framework
2. FBSDKCoreKit.framework
3. FBSDKLoginKit.framework
4. FBSDKShareKit.framework
5. Bolts.framework
6. Security.framework
7. SystemConfiguration.framework
8. GameKit.framework(게임 센터 사용 시 импорт)
9. extension 폴더 내의 res.bundle, Views.storyboard
10. KakaoGameSDK.framework
11. KakaoOpenSDK.framework
12. NaverCafeSDK 폴더 내에 있는 모든 파일

1) Info.plist 구성

```
<key>LSRequiresiPhoneOS</key>
<true/>
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key>
<true/>
</dict>
```

Facebook 소셜 인증 사용 시

```
<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>fb앱아이디</string>
    </array>
  </dict>
</array>

<key>FacebookAppID</key>
<string>앱아이디</string>
<key>FacebookDisplayName</key>
<string>앱이름</string>
<key>LSApplicationQueriesSchemes</key>
<array>
  <string>fbauth2</string>
</array>
```

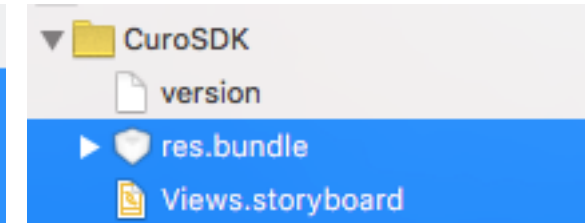
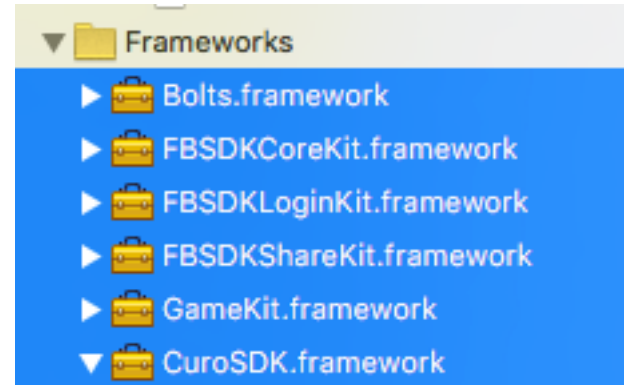
Facebook 소셜 인증 사용 시(계속)

```
<key>LSRequiresiPhoneOS</key>
<true/>
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
  <key>NSExceptionDomains</key>
  <dict>
    <key>akamaihd.net</key>
    <dict>
      <key>NSIncludesSubdomains</key>
      <true/>
      <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
      <false/>
    </dict>
    <key>facebook.com</key>
    <dict>
      <key>NSIncludesSubdomains</key>
      <true/>
      <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
      <false/>
    </dict>
    <key>fbcdn.net</key>
    <dict>
      <key>NSIncludesSubdomains</key>
      <true/>
      <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
      <false/>
    </dict>
  </dict>
</dict>
```

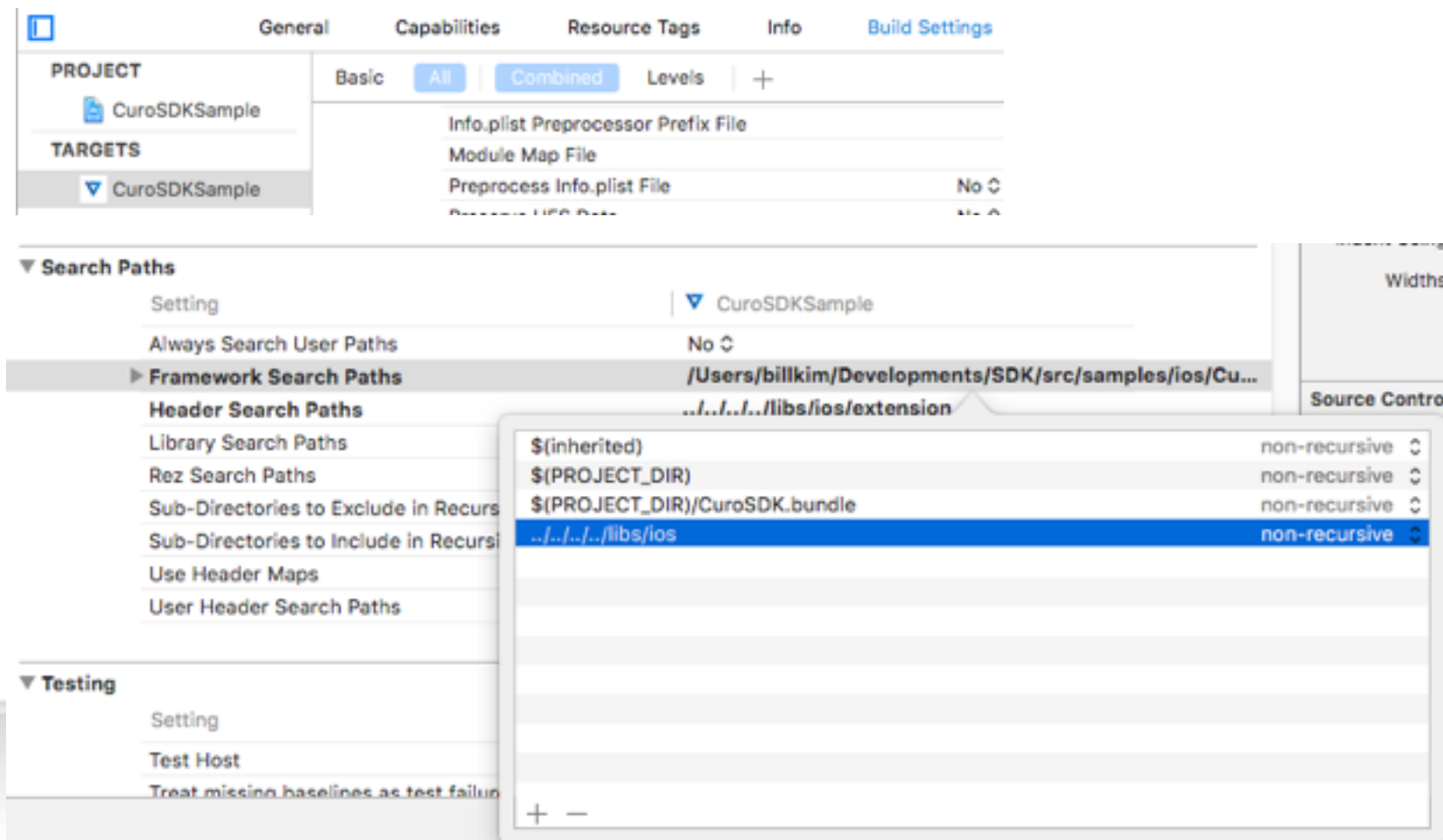

2) 아래와 같이 Xcode 프로젝트에 파일을 임포트한다.

아래의 파일들을 Xcode 프로젝트내에 임포트 시킨다.

1. CuroSDK.framework
2. FBSDKCoreKit.framework
3. FBSDKLoginKit.framework
4. FBSDKShareKit.framework
5. Bolts.framework
6. Security.framework
7. SystemConfiguration.framework
8. GameKit.framework(게임 센터 사용 시 임포트)
9. extension 폴더 내의 res.bundle, Views.storyboard
10. KakaoGameSDK.framework
11. KakaoOpenSDK.framework
12. NaverCafeSDK 폴더 내에 있는 모든 파일



임포트 후에는 **Targets/Build Settings/Framework Search Paths** 설정이 현재 프레임워크 파일이 있는 곳을 제대로 링크하고 있는지 확인한다.



1) AppDelegate 의 didFinishLaunchingWithOptions 에서 초기화 코드 적용(필수)

```

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // SDK didFinishLaunchingWithOptions 앱 델리게이트 연결
    [gCRSDKManager didFinishLaunching:application launchOptions:launchOptions];

    // SDK 기본 초기화
    [gCRSDKManager initWithGameID:@"10000"                                // VOLTUS 플랫폼 게임 아이디
                                     mode:CRServiceModeBeta                // 서비스 모드
                                     rootViewController>window.rootViewController // 메인 뷰 컨트롤러
                                     delegate:self                        // 앱 델리게이트
                                     gamecenter:YES                      // 게임 센터 사용 여부
                                     itemUrl:@"URL"];                     // 무료 충전소 관련 아이템 지급 서버 URL

    // 푸시 알림 초기화
    [gCRSDKManager initPush];

    return YES;
}

```

2) 기본 화면 방향 설정

게임에서의 기본 화면 방향(Orientation)을 설정한다. 설정이 없을 시에는 기본적으로 전체 화면 방향(CRScreenOrientationAll)으로 지정된다. 아래의 함수를 각 게임의 화면에 맞춰 초기화 함수 선언 이후에 선언하도록 한다.

```
[gCRSDKManager setScreenOrientation:CRScreenOrientationLandscape];
```

화면 방향 종류

Type	Description
CRScreenOrientationAll	모든 화면 방향
CRScreenOrientationPortrait	세로 화면 방향
CRScreenOrientationLandscape	가로 화면 방향

1) AppDelegate.h 내에 CRSDKManagerDelegate 설정(필수)

```
#import <CuroSDK/CuroSDK.h>

@interface AppDelegate : UIResponder <UIApplicationDelegate, CRSDKManagerDelegate>
```

2) AppDelegate 의 openURL 코드 적용(필수)

```
- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:(NSString *)sourceApplication annotation:(id)annotation
{
    return [gCRSDKManager application:application openURL:url sourceApplication:sourceApplication annotation:annotation];
}
```

3) 푸시 알림 관련 앱 델리게이트 코드 적용(옵션)

```
- (void)application:(UIApplication *)application didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken
{
    [gCRSDKManager application:application didRegisterForRemoteNotificationsWithDeviceToken:deviceToken];
}

- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo
{
    [gCRSDKManager application:application didReceiveRemoteNotification:userInfo];
}
```

1) 각종 이벤트 델리게이트 코드 적용

AppDelegate.cpp 파일 내에 아래의 각종 이벤트 발생에 대한 델리게이트 코드를 적용한다.

```

- (void)didFinishedLogin
{
    NSLog(@"didFinishedLogin"); // 필수
}

- (void)didFailedLogin:(CRResponseCode)code
{
    NSLog(@"didFailedLogin"); // 필수
}

- (void)didFinishedLogout
{
    NSLog(@"didFailedLogin - code : %d", code); // 옵션
}

- (void)didFailedLogout
{
    NSLog(@"didFailedLogout"); // 옵션
}

- (void)didNotConnectedNetwork
{
    NSLog(@"didNotConnectedNetwork"); // 옵션
}

- (void)didReceiveLinkType:(NSString *)type
{
    NSLog(@"didReceiveLinkType : %@", type); // 옵션
}

- (void)didReceivePushUrl:(NSString *)uri
{
    NSLog(@"didReceivePushUrl : %@", uri); // 옵션
}

- (void)didAgreedTerms
{
    NSLog(@"didAgreedTerms"); // 옵션
}

- (void)didFinishedBilling:(CRBillingState)state
{
    NSLog(@"didFinishedBilling : %d", state); // 옵션
}

- (void)didReceivedResponseCode:(CRResponseCode)code
{
    NSLog(@"didReceivedResponseCode : %d", code); // 옵션
}

- (void)didFinishedBannerPopup:(NSString *)type
{
    NSLog(@"didFinishedBannerPopup : %@", type); // 옵션
}

```

이벤트 델리게이트 목록

델리게이트 메소드	설명
didFinishedShowLogo	플랫폼 로고가 등장하고 끝났을 때 호출되는 델리게이트 함수
didFinishedLogin	로그인 성공 시에 호출되는 델리게이트 함수
didFailedLogin	로그인 실패 시에 호출되는 델리게이트 함수(자세한 코드에 대해서는 5.3 코드 목록표 참조. 29페이지)
didAgreedTerms	약관을 모두 동의할 시에 호출되는 델리게이트 함수
didFinishedLogout	로그아웃 성공 시에 호출되는 델리게이트 함수
didFailedLogout	로그아웃 실패 시에 호출되는 델리게이트 함수
didNotConnectedNetwork	네트워크 연결 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 델리게이트 함수, web 타입은 호출되지 않는다.
didFinishedBilling	인앱 구매 성공 및 실패 여부를 알려주는 델리게이트 함수(5.3 코드 목록표 참조)
didReceivedResponseCode	쿠폰 등록 및 기타 코드 값을 수신 시 호출되는 델리게이트 함수(CRResponseCode 참조, 5.3 코드 목록표 참조)
didFinishedBannerPopup	모든 배너 출력이 완료되면 호출되는 델리게이트 함수(5.3 코드 목록표 참조)

소셜 인증 기능

VOLTUS 플랫폼은 각종 소셜 인증 기능을 지원한다. iOS의 경우는 **CRPlatformTypeGameCeneter** 타입을 기본 플랫폼 인증으로 사용한다.

버전 별 소셜 인증 기능 지원 목록은 아래와 같다. 게임 센터(**CRPlatformTypeGameCeneter**)의 경우에는 로그인 성공 후 VOLTUS 플랫폼 서버로의 로그인을 시도하며 그외의 SNS(Facebook, Google+ 등) 타입은 각 SNS 인증만 진행하며 별도의 플랫폼 서버의 로그인도 시도하지 않는다.

게임 센터 로그인이 성공하면 단말기의 **UUID**와 게임 **ID**를 단말기 내에 영구적으로 저장하고 있어 다음 게임 실행 시에 게임 센터 로그인이 실패하더라도 기존의 게임 ID를 기반으로 게임 센터 로그인을 자동으로 한다. 만일 게임 센터 로그인이 단말기에서 한번도 한 이력이 없고 게임 센터 로그인 실패하면 내부적으로 게스트 로그인으로 로그인을 시도한다.

버전	지원 소셜 인증
0.9.3	Game Center, 기존 Facebook, Google+ 등의 인증은 SNS 인증만 하며 플랫폼 로그인도 하지 않도록 변경
0.9.1	Facebook, Google+, Guest
0.8.0	Facebook, Guest

로그인 팝업 API

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

```
[gCRSDKManager showLoginPopup];
```

로그인 API(별도 팝업 출력 X)

해당 함수 호출 시 원하는 플랫폼에 대해서 별도의 팝업없이 로그인 과정을 진행한다.

```
[gCRSDKManager login:CRPlatformTypeGameCenter];
```

Parameters

```
typedef enum CRPlatformType
{
    CRPlatformTypeNone = -1,
    CRPlatformTypeGuest = 100,
    CRPlatformTypeGameCenter = 102,
    CRPlatformTypeFacebook = 0,
    CRPlatformTypeGooglePlus,
}CRPlatformType;
```

로그아웃 API

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

```
[gCRSDKManager logout:^(CRResponseType type)
{
}];
```

Response

```
typedef enum CRResponseType
{
    CRResponseTypeOK = 0,
    CRResponseTypeError
}CRResponseType;
```

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.

```
[gCRSDKManager verifyLogin];
```

Return

BOOL

YES : 로그인 되어 있을 경우
NO : 로그인이 되어 있지 않을 경우

약관 팝업 출력 API

해당 함수 호출 시 기본 플랫폼 약관 팝업을 출력한다.

```
[gCRSDKManager showTermsPopup:YES];
```

Parameters

BOOL

YES : 강제로 팝업 출력

NO : 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. 공지 팝업 내에는 '무료 충전소' 기능이 포함되어 있다.

```
[gCRSDKManager showBannerPopup:BANNER_TYPE_LOGIN];
```

Parameters

NSString *

BANNER_TYPE_LOGIN : 로그인 팝업
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.

```
[gCRSDKManager showNoticeListPopup:NO];
```

Parameters

BOOL

YES : 강제로 팝업 출력

NO : 하루 보지 않기를 선택했다면 미출력

푸시 알림 초기화 API

해당 함수 호출 시 푸시 설정과 관련하여 초기화 작업을 진행한다.

```
[gCRSDKManager initPush];
```

푸시 알림 델리게이트 API

푸시 알림과 관련하여 디바이스 등록을 위하여 아래의 앱 델리게이트 함수 코드를 적용한다.

```

- (void)application:(UIApplication *)application didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken
{
    [gCRSDKManager application:application didRegisterForRemoteNotificationsWithDeviceToken:deviceToken];
}

- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo
{
    [gCRSDKManager application:application didReceiveRemoteNotification:userInfo];
}

```

푸시 알림으로 앱 실행 시 API

푸시 알림 수신 시 액션 타입이 “app”으로 설정 시에는 푸시 알림 터치로 앱 기동 시에는 해당 델리게이트 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

```

- (void)didReceivePushUrl:(NSString *)uri
{
    NSLog(@"didReceivePushUrl : %@", uri); // 옵션
}

```

게임 센터 활성화

게임 센터를 활성화 하기 위해서는 CRSDKManager 초기화 함수 호출 시 gamecenter 값을 YES 설정하면 된다.

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [gCRSDKManager initWithGameID:@"10000" // VOLTUS 플랫폼 게임 아이디
                      mode:CRServiceModeBeta // 서비스 모드
                      rootViewController>window.rootViewController // 메인 뷰 컨트롤러
                      delegate:self // 앱 델리게이트
                      gamecenter:YES]; // 게임 센터 사용 여부

    return YES;
}
```

업적 관련 API

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

```
// 업적 달성 팝업 출력
[gCRSDKManager openAchievementUI];

// 업적 오픈 - 업적 아이디를 입력하면 해당 업적을 오픈할 수 있다.
[gCRSDKManager unlockAchievement:@"업적아이디"];
```

리더보드 관련 API

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

```
// 리더 보드 팝업을 출력한다. - 대표 설정 아이디를 입력하면
// 리더 보드 메인 화면이 출력된다.
[gCRSDKManager openLeaderboardUI:@"리더보드 대표아이디"];

// 리더 보드 점수 등록
[gCRSDKManager submitScoreToLeaderboard:@"리더보드 아이디" score:@"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, didFinishBilling 이벤트 델리게이트 수신 후 대기 화면을 종료한다.

```
[gCRSDKManager purchaseItem:@"상품아이디" txID:@"플랫폼 서버에서 받은 TXID" itemUrl:@"아이템 지급요청을 호출할 서비스 서버 경로"];
```

Parameters

NSString *productID

: 스토어에 등록된 상품 아이디

NSString *txID

: 플랫폼 서버에서 구매 예약 시 발급받은 txID

NSString *itemUrl

: 아이템 지급 요청을 호출할 서비스 서버 경로 URL

플랫폼 로고 출력 API

해당 함수를 호출하여 플랫폼 로고를 출력한다. 보통 앱 시작 처음에 해당 로고를 보여준다.

```
[gCRSDKManager showLogoPopup:NO];
```

Parameters

BOOL fixedRotation

: 단말기 회전을 고정하는 매개변수, NO로 설정 시 단말기가 회전을 하여도 플랫폼 로고의 방향이 변하지 않는다.

쿠폰 등록 API

해당 함수를 호출하여 쿠폰을 등록을 요청한다.

등록 요청 후 didReceivedResponseCode 이벤트 리스너를 통하여 쿠폰 등록에 관한 코드를 받아온다. 코드 값을 0을 수신 시 정상 등록이 된 경우이다.

```
[gCRSDKManager registerCoupon:@"쿠폰 코드" itemUrl:"쿠폰 등록 요청을 호출할 서비스 서버 경로"];
```

Parameters

NSString *couponCode

: 등록 요청할 쿠폰 코드

NSString *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 경우에는 아래의 함수를 호출하여 무료 충전소 화면을 출력할 수 있다.

```
[gCRSDKManager showFreeChargingPopup];
```

무료 충전소 아이템 서버 설정 API

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

```
[gCRSDKManager showFreeChargingPopup:@"아이템 지급 서버 URL"];
```

Parameters

NSString* url

: 아이템을 지급하는 서버의 URL 주소

푸시 수신 동의 설정 API

푸시 수신 동의 여부를 정할 수 있는 API 함수이다. 해당 함수로 YES로 하면 푸시 수신을 동의하게 되며, NO로 하게 되면 수신을 거부하여 해당 유저에게는 푸시가 전송이 되지 않는다. 해당 함수 호출 후 **isSuccess**가 YES로 오면 수정이 성공한 경우이다.

```
[gCRSDKManager modifyPushPermission:YES handler:^(BOOL isSuccess)
{
    if(isSuccess)
    {
        NSLog(@"modifyPushPermission is succeeded.");
    }
    else
    {
        NSLog(@"modifyPushPermission is failed.");
    }
}];
```

Parameters

BOOL allow

: 푸시 수신 동의 여부. 수신 동의 시 YES

야간 푸시 수신 동의 설정 API

야간 푸시 수신 동의 여부를 정할 수 있는 API 함수이다. 해당 함수로 YES로 하면 야간 푸시 수신을 동의하게 되며, NO로 하게 되면 수신을 거부하여 해당 유저에게는 야간 푸시가 전송이 되지 않는다. 해당 함수 호출 후 **isSuccess**가 YES로 오면 수정이 성공한 경우이다.

```
[gCRSDKManager modifyNightPushPermission:YES handler:^(BOOL isSuccess)
{
    if(isSuccess)
    {
        NSLog(@"modifyNightPushPermission is succeeded.");
    }
    else
    {
        NSLog(@"modifyNightPushPermission is failed.");
    }
}];
```

Parameters

BOOL allow

: 야간 푸시 수신 동의 여부. 수신 동의 시 YES

Naver Cafe Plug SDK 연동

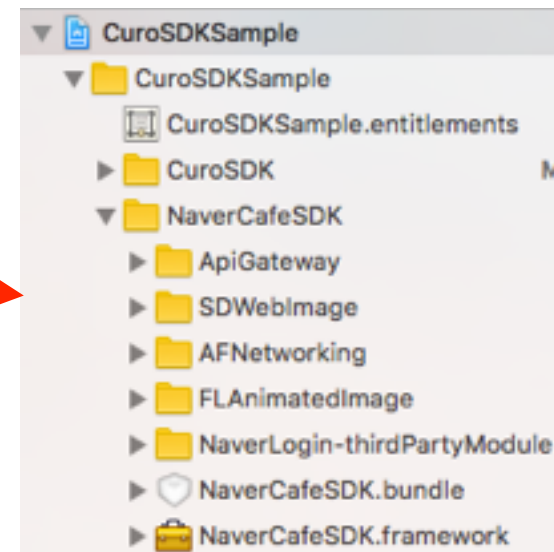
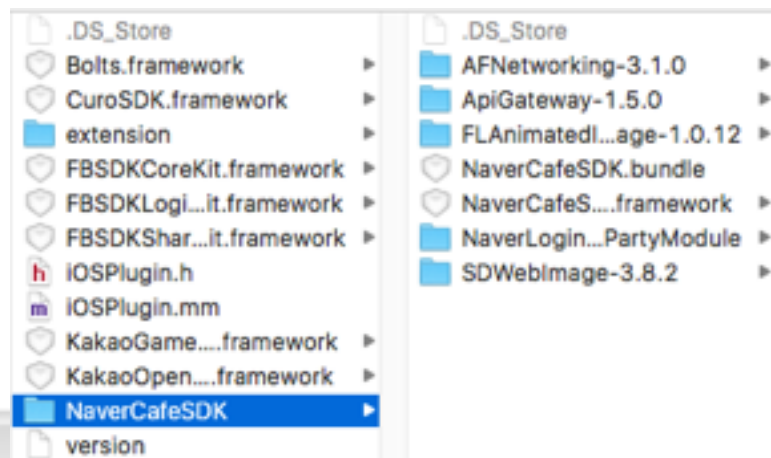
Naver Cafe PLUG SDK는 Naver에서 제작한 모바일 게임에서 이탈하지 않고 커뮤니케이션이 가능한 IN-GAME COMMUNITY 라이브러리이다.
Naver Cafe PLUG SDK는 손쉽게 커뮤니케이션 기능을 게임 안에 넣을 수 있다.

Info.plist 설정

Info.plist 파일 안에 **NSPhotoLibraryUsageDescription** 키를 추가한다.

Xcode 프로젝트 설정

1. Build Settings > Other Linker Flags에 -ObjC 옵션을 추가한다.
2. Build Phases > Link Binary With Libraries에 다음 라이브러리를 추가한다.
 - MobileCoreServices.framework (Required)
 - SystemConfiguration.framework (Required)
 - MediaPlayer.framework (Required)
 - AVFoundation.framework (Required)
 - CoreMedia.framework (Required)
 - AssetsLibrary.framework (Required)
 - Security.framework (Required)
 - ImageIO.framework (Required)
 - QuartzCore.framework (Required)
 - ReplayKit.framework (**Optional**)
3. VOLTUS SDK 내에서 NaverCafeSDK 폴더 안에 파일들을 아래와 같이 Xcode 프로젝트에 포함시킨다.



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.

```
[gCRSDKManager initNaverCafeSDK:@"197CymaStozo7X5r2qR5" clientSecret:@"evCgKH1kJL" cafeId:28290504
landscapeMode:YES];
```

Parameters

```
NSString *clientId
: 클라이언트 아이디

NSString *clientSecret
: 클라이언트 Secret 값

NSInteger cafeId
: 카페 아이디

BOOL landscapeMode
: 가로 모드 인지 체크하는 변수
```

초기화 API(글로벌 채널용)

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

```
[gCRSDKManager initNaverCafeSDKGlobal:@"197CymaStozo7X5r2qR5" cafeId:28290504 landscapeMode:YES];
```

Parameters

```
NSString *consumerKey
: 클라이언트 아이디

NSInteger cafeId
: 카페 아이디

BOOL landscapeMode
: 가로 모드 인지 체크하는 변수
```

초기화 API(글로벌 채널용)

화면에 Naver Cafe 위젯(Widget) 아이콘을 표시할지 말지에 대하여 설정하는 함수이다.

아래의 함수에서 YES 로 설정하여 호출하면 화면 측면에 위젯 아이콘이 표시되어 카페 커뮤니티 접근 및 게시글, 화면 캡처, 영상 녹화 등의 기능을 사용할 수 있다.

```
[gCRSDKManager showNaverCafeWidget:YES];
```

Parameters

BOOL isShow
: Widget 을 표시할 경우 YES 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.

담당 부서 : 플랫폼 서비스실

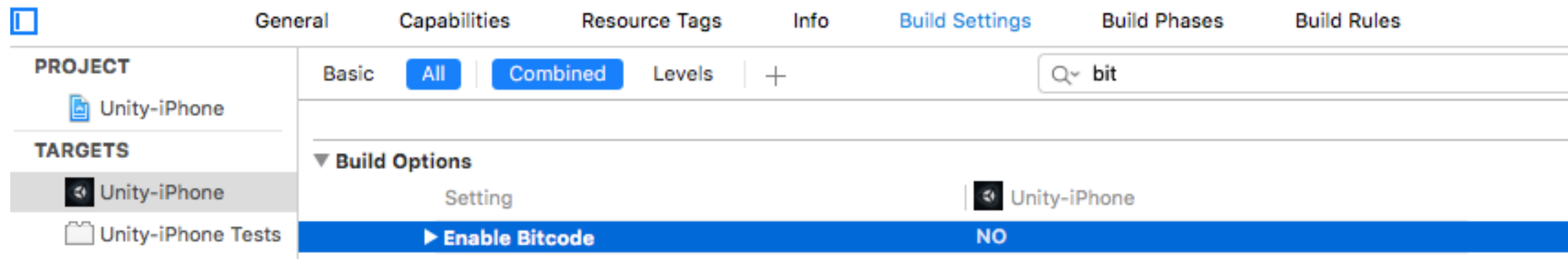
담당자 : 김정훈

이메일 : billkim@curoholdings.com

Bitcode 컴파일 에러

아래와 같이 bitcode와 관련한 컴파일 에러가 발생 시에는 Build Settings에서 Enable Bitcode를 NO로 설정 바랍니다.

```
ld: '/Users/billkim/Desktop/Voltus SDK Sample/iOS/Frameworks/Plugins/iOS/CuroSDK.framework/CuroSDK(CRAPIManager.o)' does not contain bitcode. You must rebuild it with bitcode enabled (Xcode setting ENABLE_BITCODE), obtain an updated library from the vendor, or disable bitcode for this target. for architecture armv7
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```



KeychainItemWrapper Duplicate symbol 에러

아래와 같이 **KeychainItemWrapper** 파일과 관련하여 중복 에러 시에는 현재 프로젝트에서 **KeychainItemWrapper.h**, **KeychainItemWrapper.m** 파일을 다른 곳에서 사용하고 있는지 체크한다. 기본적으로 VOLTUS SDK에서도 키체인 기능 활용을 위하여 KeychainItemWrapper 파일들을 사용하고 있다. 따라서 **기존의 KeychainItemWrapper.h, KeychainItemWrapper.m 파일을 사용하고 있다면 삭제하고 컴파일하길 바랍니다.**

```
duplicate symbol _OBJC_CLASS_$_KeychainItemWrapper in:
  /Users/.../Release-iphones/.../arm7/KeychainItemWrapper.o
  /Users/.../Components/Worklight/WorklightSDK/libWorklightStaticLibProject.a(KeychainItemWrapper.o)
ld: 5 duplicate symbols for architecture arm7
```

코드 목록표(CRConfig.h 참조)

didFinishedBilling 이벤트 수신 시 받을 수 있는 code 목록표

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

didReceivedResponseCode 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)

didFailedLogin 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

CRUser 주요 변수

유저의 정보를 담고있는 객체

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