

PATIGAMES Publish SDK v1.4.1

(for iOS)

이 문서는 Android 버전과 비교해 코드/설정의 차이점만 기술되어있으므로 반드시 Android 버전과 함께 확인할 것! (주의 사항, 필요한 추가구현 등)

API 사용 전 협의 사항

파티게임즈에서 제공하는 관리페이지에 게임이 등록되어 있는지, SDK/API 메뉴에서 gameId / ClientData (manifest에 추가하는 encryptedClientData) / serverKey등이 발급되어 있는지 확인한다.

관리 페이지 : <https://bo.patigames.com> (라이브) / <https://botest.patigames.com> (스테이지)

Library 추가

[공통]

1. 아래 라이브러리/프레임워크들을 프로젝트 빌드 타겟의 Build Phases - Link Binary With Libraries에 추가한다.

```
- libPatiSDK-armv7-armv7s-arm64-i386.a
- StoreKit.framework
- Security.framework
- AdSupport.framework
- CoreGraphics.framework
- UIKit.framework
- CoreTelephony.framework
```

2. include 경로의 헤더 파일들을 프로젝트에 포함시킨다.
3. UIApplicationDelegate를 구현하는 interface에서 다음 코드를 추가한다.

Appdelegate 설정
<pre>(Objective-C) #import <PatiPublishSDK.h> - (BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:(NSString *)sourceApplication annotation:(id)annotation { return PatiSDK::iOS::handleOpenURL(url, sourceApplication, annotation); }</pre>

4. Info.plist를 열고, 아래와 같은 값을 추가한다.

Key	Type	Value
PatiEncryptedClientData	String	발급받은 ClientData
PatiUsingPrivacyInfo	Boolean	게임의 개인 정보 수집 여부 (없으면 false)

5. Build Settings - Linking - Other Linker Flags에 **-ObjC** 를 추가한다.

* STL 관련 링크 오류가 발생할 경우, Build Settings - Language C++에서 C++ Language Dialect를 C++11 [-std=c++11]로, C++ Standard Library를 libc++로 변경해본다.

** -ObjC Flag를 추가한 뒤, 링크시 GameController 관련 오류가 발생하면 GameController.framework를 Build Phases - Link Binary With Libraries에 추가한다.

[카카오톡]

0. 위의 공통 작업을 선행한다.

1. 아래 라이브러리를 프로젝트에 추가하고, 프로젝트 빌드 타겟의 Build Phases - Link Binary With Libraries에 추가한다.

```
- libPatiSDKKakao-armv7-armv7s-arm64-i386.a  
- external/libkakao-ios-sdk-i386-armv7-armv7s-arm64.a
```

2. URL types에 새로운 값을 추가하고, identifier에 앱의 BundleID로, URL schemes에 "kakao%{*kakaoClientID*}"를 넣는다. 이 작업은 Info.plist를 직접 수정해도 되고, XCode에서 지원하는 편집 기능을 사용해도 된다.

3. Build Settings - Linking - Other Linker Flags에 **-all_load** 를 추가한다.

4. Build Settings - Custom Compiler Flags에 -DPATISDK_KAKAOTALK을 추가한다.

5. initSDK가 호출된 후에 아래 코드를 같이 호출한다.

초기화 - Kakao

(Objective-C)

```
PatiSDK::initKakao(<const char *clientId>, <const char *clientSecret>);
```

[페이스북]

0. 위의 공통 작업을 선행한다.

1. 아래 라이브러리/프레임워크를 프로젝트에 추가하고, 프로젝트 빌드 타겟의 Build Phases - Link Binary With Libraries에 추가한다.

```
- libPatiSDKFacebook-armv7-armv7s-arm64-i386.a  
- external/FacebookSDK.framework
```

2. Info.plist를 열고, 아래와 같은 값을 추가한다.

Key	Type	Value
FacebookAppId	String	페이스북 App ID
FacebookDisplayName	String	페이스북 Display Name (App Dashboard에 있음)

3. URL types에 새로운 값을 추가하고, identifier에 앱의 BundleID로, URL schemes에 앞에 "fb"문자열을 붙인 페이스북 App ID를 넣는다. 이 작업은 Info.plist를 직접 수정해도 되고, XCode에서 지원하는 편집 기능을 사용해도 된다.

4. Build Settings - Custom Compiler Flags에 -DPATISDK_FACEBOOK을 추가한다.

5. UIApplicationDelegate를 구현하는 interface에서 다음 코드를 추가한다.

Appdelegate 설정
<pre>(Objective-C) #import <FacebookSDK/FacebookSDK.h> - (void)applicationDidBecomeActive:(UIApplication *)application { [FBAppCall handleDidBecomeActive]; }</pre>

6. initWithSDK가 호출된 후에 아래 코드를 같이 호출한다.

초기화 - Facebook
<pre>(Objective-C) PatiSDK::initFacebook();</pre>

[구글+]

0. 위의 공통 작업을 선행한다.

1. 아래 라이브러리/프레임워크를 프로젝트에 추가하고, 프로젝트 빌드 타겟의 Build Phases - Link Binary With Libraries 에 추가한다.

```
- libPatiSDKGooglePlus-armv7-armv7s-arm64-i386.a
- external/GooglePlus.framework
- external/GoogleOpenSource.framework
- AddressBook.framework
- AssetsLibrary.framework
- Foundation.framework
- CoreLocation.framework
- CoreMotion.framework
- CoreGraphics.framework
- CoreText.framework
- MediaPlayer.framework
- SystemConfiguration.framework
- UIKit.framework
```

2. Info.plist를 열고, 아래와 같은 값을 추가한다.

Key	Type	Value
GooglePlusClientId	String	구글+ 개발자 콘솔에서 발급받은 Client ID
GooglePlusScopes	Array	로그인시 추가로 필요한 Scope의 배열. 이 값을 추가하지 않아도 기본으로 plus.login, userinfo.profile을 조회하며, 배열에 String으로 값을 추가해서 추가로 설정할수있다. 입력가능한 Scope들은 링크를 참조할 것. (https://developers.google.com/oauthplayground/)

3. URL types 에 새로운 값을 추가하고, identifier 에 앱의 BundleID 로, URL schemes 에도 앱의 BundleID 를 넣는다. 이 작업은 Info.plist 를 직접 수정해도 되고, XCode 에서 지원하는 편집 기능을 사용해도 된다.

4. Build Settings - Custom Compiler Flags 에 -DPATISDK_GOOGLEPLUS 을 추가한다.

5. initWithSDK가 호출된 후에 아래 코드를 같이 호출한다.

초기화 - Google+

(Objective-C)

```
PatiSDK::initGooglePlus();
```

6. Custom Scope 를 이용할 경우(구글 플러스 이외의 구글 API 사용), 각 API 에 맞는 Manager 를 직접 signin 하여 사용해야 한다. (ex - Google Play Game Services API 를 이용하기 위해서는 scope 에 <https://www.googleapis.com/auth/games> 를 추가하고 GPGManager 를 직접 signin)

[네이버]

0. 위의 공통 작업을 선행한다.

1. 아래 라이브러리/프레임워크를 프로젝트에 추가하고, 프로젝트 빌드 타겟의 Build Phases - Link Binary With Libraries 에 추가한다.

```
- libPatiSDKNaver-armv7-armv7s-arm64-i386.a  
- external/libNaverLogin.a  
- libxml2.dylib
```

2. URL types 에 새로운 값을 추가하고, identifier 에 앱의 BundleID 로, URL schemes 에 앞에 네이버 개발자 센터에서 설정한 URL Scheme 값을 넣는다. 이 작업은 Info.plist 를 직접 수정해도 되고, XCode 에서 지원하는 편집 기능을 사용해도 된다.

3. initWithSDK가 호출된 후에 아래 코드를 같이 호출한다.

초기화 - Naver

(Objective-C)

```
PatiSDK::initNaver(<const char* clientID>, <const char* clientSecret>,  
                  <const char* appName>, <const char* urlScheme>  
);
```

clientID, clientSecret : 네이버 개발자 센터에서 확인할 수 있다.

appName : 서비스 앱의 이름

urlScheme : 네이버 로그인 프로세스가 완료된 후 응답받을 게임의 URL Scheme (2 번에서 설정한 값)

푸시 수신 설정

[초기화]

푸시 수신을 하려면 먼저 APNS에 등록을 해야한다. iOS로부터 devicetoken을 받아오기 위해 AppDelegate의 application:didRegisterForRemoteNotificationsWithDeviceToken를 아래와 같이 구현한다.

iOS DeviceToken 설정 예제
<pre>(Objective-C) - (void)application:(UIApplication *)application didRegisterForRemoteNotificationsWithDeviceToken:(NSData*)deviceToken { PatiSDK::iOS::registerPushToken(deviceToken); }</pre>

(* 유니티의 경우는 유니티 문서의 푸시 서비스 내용 참조 바람.)

didRegisterForRemoteNotificationsWithDeviceToken은 사용자가 푸시 수신 동의 메시지에 동의해야 실행되며, 거부했거나 APNS 등록에 실패한 경우에는 application:didFailToRegisterForRemoteNotificationsWithError가 호출되므로, 필요에 따라 코드를 추가한다.

푸시를 키고 끄거나, 켜져있는지 확인하는 방법은 안드로이드/유니티 문서의 코드와 동일하다.

[테스트]

최초로 실행시에는 APNS 동의를 확인하는 alert가 뜨지만 이후부터는 디바이스가 restore되거나 마지막으로 삭제한지 하루가 넘지 않으면 다시 뜨지 않는다. 만약 처음 실행한 것처럼 테스트가 필요하다면 아래처럼 하면 된다고 한다.

1. 디바이스에서 앱을 삭제한뒤 디바이스를 완전히 종료 후 재시작
2. Date&Time 설정에서 1일 이상의 시간을 앞당김
3. 디바이스를 완전히 종료한뒤 재시작, 그리고 테스트

(출처: <https://developer.apple.com/library/ios/technotes/tn2265/index.html>)

Resetting the Push Notifications... 항목 참조.)