AIQ.AWARE

iOS SDK Developer's Guide

Ver.04

Release date: May 15, 2020

Skelter Labs

Overview

스켈터랩스의 AIQ.AWARE 서비스는 초개인화(Hyper-personalization) 솔루션을 위한 API를 제공합니다. 수집한 고객의 데이터를 바탕으로 초개인화 기능을 쉽고 올바르게 사용하기 위해서는 스켈터랩스에서 제공하는 AIQ.AWARE iOS SDK가 필요합니다.

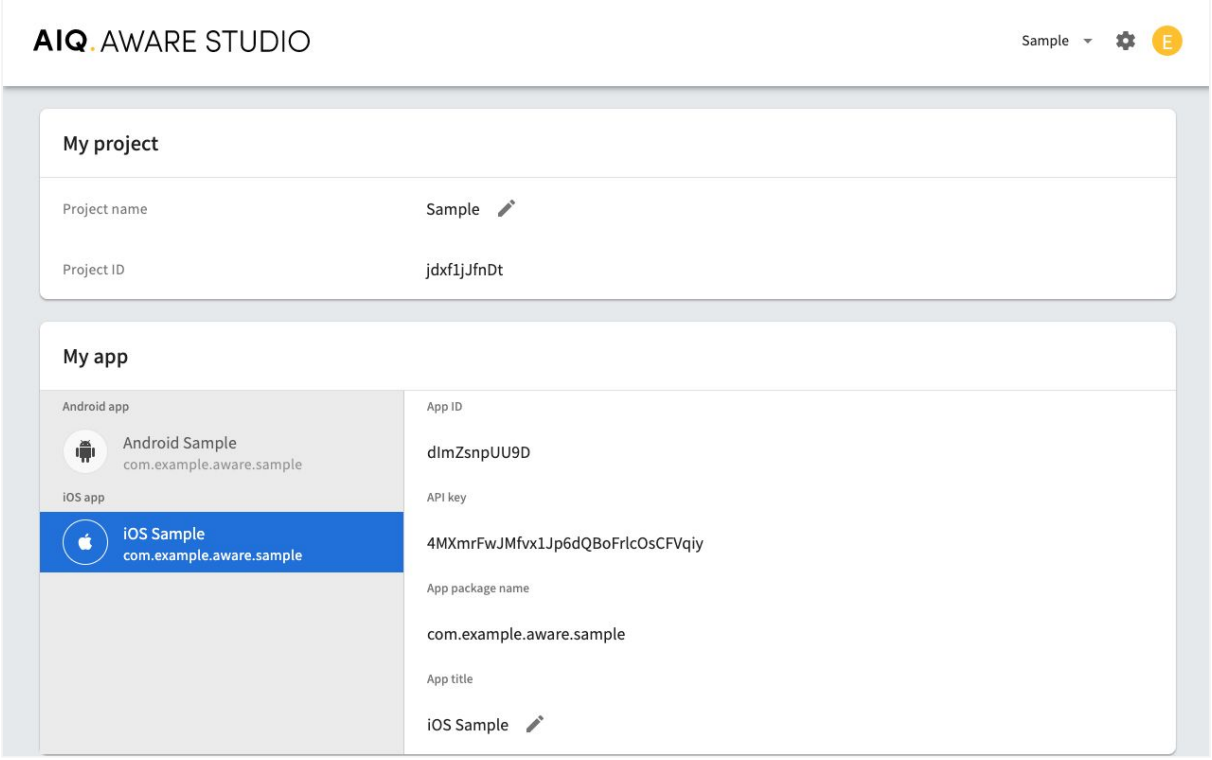
본 문서는 AIQ.AWARE 서비스를 사용하기 위한 기본 개발환경 구성 및 iOS 애플리케이션에서 필요한 SDK 사용법 뿐 아니라 AIQ.AWARE 서비스를 통해 제공되는 정보의 상세 설명을 포함하고 있습니다.

프로젝트 및 앱 생성

프로젝트 생성 요청

AIQ.AWARE iOS SDK를 사용하기 위해선 프로젝트 및 iOS 앱이 생성되어야 합니다.

프로젝트를 사용하려면 https://aware.skelterlabs.com/studio 에서 회원가입후, 신청 서베이를 통해 앱 신청을 요청하실 수 있습니다.



[프로젝트 설정 화면]

앱 사용이 승인되면 프로젝트 설정 화면에서 Apple team ID, APNs key ID, APNs auth key를 입력해주셔야 합니다. 각 값들은 Apple developer program 의 아래 항목들에서 찾아볼 수 있습니다.

Apple team ID	Membership - Team ID
APNs key ID	Certificates, Identifiers & Profiles - Keys - 사용하실 APNs Key의 Key ID
APNs auth key	상기 APNs key ID의 값(.p8 파일의 내용)

개발환경 구성

CocoaPods

AIQ.AWARE SDK는 CocoaPods를 통해 환경을 구성할 수 있습니다. 초기화 방법은 CocoaPods 문서를 참고하시기 바랍니다.

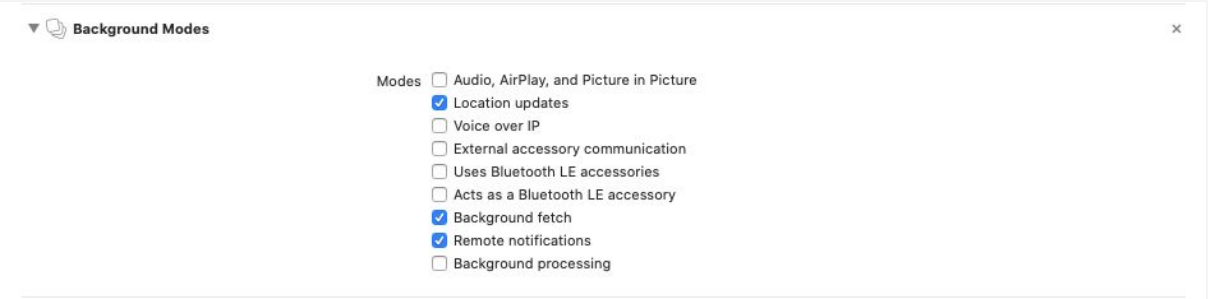
아래 설정 후 pod install을 하면 pod dependencies를 설치할 수 있습니다.

```
# Podfile
...

target 'your-app' do

  ...
  # 1. AIQ.AWARE for release
  pod 'AIQAware', :configurations => ['Release']
  # 2. AIQ.AWARE for simulator
  pod 'AIQAware-debug', :configurations => ['Debug']
  ...

end
```

권한 설정

AIQ.AWARE는 고객 정보 수집을 위해 위치, 백그라운드 작업, APNs 등을 사용합니다. 해당 기능을 사용하기 위해 프로젝트 설정의 Capabilities의 Background Modes에 권한을 추가해야 합니다.

'Location updates', 'Background fetch', 'Remote notifications'를 켜주시기 바랍니다.



[Capabilities]

Wifi 정보 수집을 위해 Capabilities에 Access WiFi information을 추가해야 합니다. (>= iOS 12)

[Capabilities]

알람을 이용한 앱 구동을 위해 Capabilities에 Push Notifications를 추가해야 합니다.

[Capabilities]

Info.plist에 정보 수집을 위해 다음 권한 string을 추가해줍니다.

- Privacy - Location When In Use Usage Description[1]
- Privacy - Location Always Usage Description
- Privacy - Location Always and When In Use Usage Description
- Privacy - Media Library Usage Description
- Privacy - Photo Library Usage Description
- Privacy - Microphone Usage Description
- Privacy - Calendars Usage Description
- Privacy - Motion Usage Description

Key	Type	Value
▼Information Property List	Dictionary	(24 items)
Privacy - Media Library Usage Description	String	Playing music information is required
Privacy - Photo Library Usage Description	String	Photo library information is required
Privacy - Microphone Usage Description	String	Microphone permission is required
Privacy - Location When In Use Usage Description	String	Will use location when you are using this app
Privacy - Location Always Usage Description	String	Will always use location
Privacy - Location Always and When In Use Usage Description	String	Will use location always and when you are usgin this app
Privacy - Calendars Usage Description	String	Calendar data is required
Privacy - Motion Usage Description	String	Motion permission is required

[Info.plist]

[1] iOS13 이상부터는 항상 이 권한을 먼저 요청하고 추후에 별도의 권한 팝업을 통해 항상 허용으로 전환하게 됩니다.

앱 설정

AIQ.AWARE SDK를 통해 고객의 정보를 수집하려면 제공되는 앱 정보를 사용해야 합니다. 앱 정보는 생성된 Studio 프로젝트 페이지를 확인하시기 바랍니다.

프로젝트에 “AwareClient-Info.plist”라는 이름의 파일을 추가하고, 앱 정보를 입력합니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
	<key>API_KEY</key>
	<string>AAAAAAAAAAAAAAAAAAAAAAAAAAAA</string>
	<key>APP_ID</key>
	<string>AAAAAAAAAA</string>
	<key>PROJECT_ID</key>
	<string>AAAAAAAAAA</string>
</dict>
</plist>
```

[AwareClient-Info.plist]

Key	Type	Value
▼ Information Property List	Dictionary	(3 items)
API_KEY	String	AAAAAAAAAAAAAAAAAAAAAAAAAAAA
APP_ID	String	AAAAAAAAAA
PROJECT_ID	String	AAAAAAAAAA

[Xcode - AwareClient-Info.plist]

앱에는 위치, 백그라운드 작업, APNs 설정을 해야 SDK를 사용할 수 있습니다.

```
// AppDelegate.swift
…
import AIQAware

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

  func application(
    _ application: UIApplication,
    didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey:
Any]?
  ) -> Bool {
    // Init AIQ.AWARE SDK
    AIQAware.configure(self)
    AIQAware.handleLaunchOptions(launchOptions)

    // Enable APNs
    application.registerForRemoteNotifications()
```

```
    // Enable background fetch
    application.setMinimumBackgroundFetchInterval(
      UIApplication.backgroundFetchIntervalMinimum)
    return true
  }

  func application(
    _ application: UIApplication,
    performFetchWithCompletionHandler completionHandler: @escaping
(UIBackgroundFetchResult) -> Void
  ) {
    // Handle background task
    AIQAware.performBackgroundFetchTask {
      completionHandler(.newData)
    }
  }

  func application(
    _ application: UIApplication,
    didRegisterForRemoteNotificationsWithDeviceToken deviceToken: Data
  ) {
    // Register APNs
    // APNs locates device by its token distinguished by runtime environment.
#if DEBUG
    AIQAware.registerSandboxRemoteNotification(
      with: deviceToken, bundleId: Bundle.main.bundleIdentifier!)
#else
    AIQAware.registerRemoteNotification(
      with: deviceToken, bundleId: Bundle.main.bundleIdentifier!)
#endif
  }


  func applicationDidEnterBackground(_ application: UIApplication) {
    // Handle background ending process
    AIQAware.applicationDidEnterBackground(application)
  }

  func application(
    _ application: UIApplication,
    didReceiveRemoteNotification userInfo: [AnyHashable: Any],
    fetchCompletionHandler completionHandler: @escaping (UIBackgroundFetchResult) ->
Void
  ) {
    // Handle AIQ.AWARE remote notifications
    if AIQAware.handleRemoteNotification(userInfo) {
      completionHandler(.newData)
      return
    }

    // Handle non-AIQ.AWARE remote notifications
  }
}


// MARK: - AIQAwareDelegate
extension AppDelegate: AIQAwareDelegate {
```

```
...
}
```

캠페인 정보 수신 설정

캠페인을 생성할 때 설정해둔 조건이 충족되었을 때 유저는 Silent Push Notification의 형태로 캠페인 정보를 받게 되며, SDK에서 자체적으로 캠페인 정보를 이용해 Notification을 표시해 줍니다. 단, 앱이 foreground에 있을 때에는 Notification을 보여줄 수 없으므로 Alert 형식으로 캠페인의 내용을 표시해 줍니다.

앱에서 Notification을 보여주기 위해서는 아래와 같이 추가적인 권한 설정이 필요합니다. 처음 한번에 한해 유저에게 푸시알림 권한 요청 팝업이 보여지게 됩니다.

```
// AppDelegate.swift

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

  func application(
    _ application: UIApplication,
    didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey:
Any]?
  ) -> Bool {
    // Additional remote notification permission handlings
    if #available(iOS 10.0, *) {
      let authOptions: UNAuthorizationOptions = [.alert]
      UNUserNotificationCenter.current().requestAuthorization(
        options: authOptions,
        completionHandler: { _, _ in
          DispatchQueue.main.async {
            application.registerForRemoteNotifications()
          }
        }
      )
      // Needed to present notifications
      UNUserNotificationCenter.current().delegate = self
    } else {
      let settings = UIUserNotificationSettings(types: [.alert], categories: nil)
      application.registerUserNotificationSettings(settings)
      application.registerForRemoteNotifications()
    }

    return true
  }
}
```

또한 캠페인 Notification을 클릭해 유저가 반응했을 때 필요한 처리를 하기 위해 아래 설정이 추가적으로 필요합니다. OS버전에 따라 적절한 지점에 명시된 SDK function을 호출해주시면 됩니다.

```
@available(iOS, introduced: 9.0, deprecated: 10.0)
extension AppDelegate {
  func application(
    _: UIApplication,
    handleActionWithIdentifier _: String?,
    forRemoteNotification userInfo: [AnyHashable: Any],
    completionHandler: @escaping () -> Void
  ) {
    defer {
      completionHandler()
    }

    if AIQAware.handleAction(forRemoteNotification: userInfo) {
      return
    }
  }
}

@available(iOS 10.0, *)
extension AppDelegate: UNUserNotificationCenterDelegate {
  func userNotificationCenter(
    _: UNUserNotificationCenter,
    didReceive response: UNNotificationResponse,
    withCompletionHandler completionHandler: @escaping () -> Void
  ) {
    defer {
      completionHandler()
    }

    if AIQAware.handleUserNotification(didReceive: response) {
      return
    }
  }
}
```

AIQ.AWARE 서비스 시작

SDK 사용 등록

SDK 사용 등록을 해야 고객 정보 수집을 시작할 수 있습니다.

```
// ViewController.swift

class ViewController: UIViewController {

  private var authManager: AIQAwareAuthManager {
    get {
      let manager = AIQAwareAuthManager.instance
      manager.authManagerDelegate = self
      return manager
    }
  }
```

```
  func register() {
    authManager.register()
  }
}

extension ViewController: AIQAwareAuthManagerDelegate {
  func registerSucceed(token: String) {
    // Handle when registration is succeed.
  }

  func registerFailed(reason: String) {
    // Handle when registration is failed.
  }
...
}
```

고객 정보 수집 시작

```
// ViewController.swift
class ViewController: UIViewController {

  func startCollect(collectorType: CollectorType) {
    AIQAwareSignalManager.instance.collect(type: collectorType)
  }

  func startCollectAirPressure() {
    startCollect(CollectorType.airPressure)
  }

  func startCollectBattery() {
    startCollect(CollectorType.battery)
  }

  func startCollectDetectedActivity() {
    startCollect(CollectorType.detectedActivity)
  }

  func startCollectGravity() {
    startCollect(CollectorType.gravity)
  }

  func startCollectLocation() {
    startCollect(CollectorType.location)
  }

  func startCollectSleep() {
    startCollect(CollectorType.sleep)
  }

  func startCollectStepCount() {
    startCollect(CollectorType.stepCount)
  }
```

```
  func startCollectWifi() {
    startCollect(CollectorType.wifi)
  }
}
```

SDK 사용 해제

사용 해제를 하면 SDK에서 더 이상 고객의 정보 수집을 하지 않습니다. 사용 등록을 하는 시점부터 새로운 고객으로 처리하기 때문에 정확한 정보 수집을 위해 꼭 필요한 경우 외에는 이미 설치된 앱의 SDK 사용 해제는 하지 마시기 바랍니다.

```
// ViewController.swift

class ViewController: UIViewController {

  private var authManager: AIQAwareAuthManager {
    get {
      let manager = AIQAwareAuthManager.instance
      manager.authManagerDelegate = self
      return manager
    }
  }

  func unregister() {
    authManager.unregister()
  }
}

extension ViewController: AIQAwareAuthManagerDelegate {
  func unregisterSucceed() {
    // Handle when unregistration is succeed.
  }

  func unregisterFailed(reason: String) {
    // Handle when unregistration is failed.
  }
...
}
```

샘플앱 실행

AIQ.AWARE 깃헙에서 샘플앱을 찾아보실 수 있습니다.
https://github.com/SkelterLabsInc/aware-ios-sdk

문제해결

Swift compiler 버전이 맞지 않는 경우

XCode11 이후 버전을 설치하며 Swift compiler 5.1을 사용하시기 바랍니다.

AIQ.AWARE SDK API

AIQAware

AIQ.AWARE SDK 사용을 위한 interface.

Functions

Name	Return	Description
configure(delegate: AIQAwareDelegate)	Void	AIQ.AWARE SDK 사용을 위해 instance를 초기화합니다.
handleLaunchOptions(launchOptions: [UIApplication.LaunchOptionsKey : Any]?)	Void	앱이 실행되었을 때 launch option에 따라 필요한 작업을 수행합니다.
handleRemoteNotification(userInfo: [AnyHashable: Any])	Bool	AIQ.AWARE 에서 보낸 remote notification을 처리합니다.
performBackgroundFetchTask(complete: (() -> Void)? = nil)	Void	Background task가 실행될 때 정보 수집을 수행합니다.
registerRemoteNotification(with deviceToken: Data, bundleId: String)	Void	Remote notification 등록 후 등록 정보를 SDK로 전달합니다.
registerSandboxRemoteNotificati on(with deviceToken: Data, bundleId: String)	Void	Sandbox용 Remote notification 등록 후 등록 정보를 SDK로 전달합니다.
applicationDidEnterBackground(application: UIApplication)	Void	앱이 백그라운드 상태로 바뀔 때 정보 수집을 수행합니다.
getProfile(	Void	고객의 profile 정보를 Dictionary 형태로

success: @escaping (Dictionary<String, Any>?) -> Void, failure: ((Error, Int?) -> Void)?)		반환합니다.

Variable setters

Name	Type	Description
signalManagerDelegate	AIQAwareSignalManagerDelegate	Signal manager delegate를 등록합니다.
eventDelegate	AIQAwareEventDelegate	Events delegate를 등록합니다.

AIQAwareDelegate

Name	Type	Description
permissionsToUse	Set<PermissionType>	사용하고자 하는 권한을 설정합니다. 관련된 collector들을 모두 사용할 수 있습니다. collectorsToUse보다 우선 반영됩니다.
collectorsToUse	Set<CollectorType>	permissionsToUse를 사용하지 않는 경우 사용하고자 하는 collector를 직접 설정합니다. permissionsToUse, collectorsToUse 모두 nil인 경우 모든 collector 타입을 사용할 수 있습니다.

AIQAwareSignalManager

Name	Return	Description
isCollectingSignal(ofType type: CollectorType)	Bool	주어진 수집 타입의 정보를 현재 수집중인지 여부를 반환합니다.
restoreActivateStates(trigger: Bool)	Void	마지막 수집하고있던 상태를 복원합니다. trigger=false인 경우 수집을 위한 준비상태로 되고, trigger=true인 경우 바로 수집을 시작합니다.
collect(enable: Bool = true, type: CollectorType)	Void	주어진 수집 타입의 정보를 수집을 시작합니다. 수집 타입에 해당하는 권한을 고객에게 받은 적이 없다면 권한 요청 팝업이 고객에게 보여지게 됩니다.

AIQAwareSignalManagerDelegate

Name	Description
handleMissingPermissionFor(collectorType: CollectorType)	고객 정보 수집에 필요한 권한이 없을 때 수집 타입과 함께 호출됩니다. 고객에게 권한을 재요청하는데 사용할 수 있습니다.

AIQAwareAuthManager

Functions

Name	Return	Description
register(userHint: String, ctoken: String)	Void	사용자 기기로 SDK를 사용 등록합니다. userHint에는 앱에서 사용자 식별(관리용) 및 추적하기 위한 ID를 지정할 수 있습니다. 관리를 원하지 않는 경우 nil 값을 입력하면 됩니다. 값을 지정하지 않는 경우 default 값은 nil입니다.
unregister()	Void	사용자 기기의 SDK 사용을 해제합니다.

Variables

Name	Type	Description
isRegistered	Bool	SDK 사용 등록이 되었는지 여부를 반환합니다.

AIQAwareAuthManagerDelegate

Name	Description
registerSucceed(token: String)	사용 등록 성공시 AIQ.AWARE API에 사용될 토큰과 함께 호출됩니다.
registerFailed(reason: String)	사용 등록 실패시 이유와 함께 호출됩니다.
unregisterSucceed()	사용 해제 성공시 호출됩니다.
unregisterFailed(reason: String)	사용 해제 실패시 이유와 함께 호출됩니다.

CollectorType

Name	Description
airPressure	기압 정보 수집을 위해 사용합니다.
battery	배터리 정보 수집을 위해 사용합니다.
location	위치 정보 수집을 위해 사용합니다.
sleep	수면 상태 측정을 위해 사용합니다.
wifi	와이파이 상태 정보 수집을 위해 사용합니다.

AIQAwareEventsDelegate

Name	Description
activityEventsDetected(events: [[String: Any]])	상태 변경 이벤트 발생을 수신할 때마다 시간 역순으로 정렬된 이벤트 리스트와 함께 호출됩니다.
func campaignReceived(campaign: [String: Any])	캠페인을 수신했을 때 받은 캠페인 정보와 함께 호출됩니다.

이벤트

상태변경 이벤트

SDK가 유저의 상태 변경 이벤트를 탐지한 경우, AIQAwareEventsDelegate의 activityEventsDetected에 해당 내용을 전달합니다.

Apple의 Background Update 가이드라인에 따라 해당 업데이트는 실시간이 아닐 수 있습니다.

```
// AppDelegate.swift
...
import AIQAware


// MARK: - AIQAwareEventDelegate
extension AppDelegate: AIQAwareEventsDelegate {
  func activityEventsDetected(events: Array<Dictionary<String, Any>>) {
```

```
    // Handle events
  }
}
```

상태 변경 이벤트 데이터의 자세한 형식은 'AIQ.AWARE API document'를 참고하시기 바랍니다.

캠페인

SDK에서 캠페인을 수신하면 Notification을 표시함과 동시에 AIQAwareEventsDelegate의 campaignReceived에 해당 내용을 전달합니다.

Apple의 Background Update 가이드라인에 따라 해당 업데이트는 실시간이 아닐 수 있습니다.

```
// AppDelegate.swift
...
import AIQAware


// MARK: - AIQAwareEventDelegate
extension AppDelegate: AIQAwareEventsDelegate {
  func campaignReceived(campaign: [String: Any]) {
    // Handle Campaign
  }
}
```

기타 문의

AIQ.AWARE iOS SDK에 대해 추가적으로 문의사항이 있다면 aware-support@skelterlabs.com 으로 주시기 바랍니다.