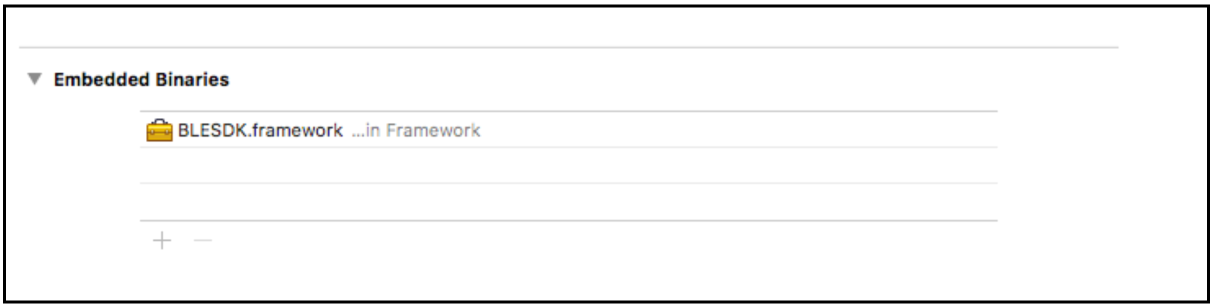
Airyzone SDK 版本：v0.0531.1

支援 iOS 版本：ios 9.0 以上

支援程式語言：Swift 3.0 以上

一、設定 Project

1.BLESDK.framework 置入 Project，並且加入至 Embedded Binaries：



▼ Embedded Binaries

BLESDK.framework ...in Framework

2.加入相依的 Frameworks， 或使用 Cocoapods

a. AFNetworking

b. Alamofire

c. CoreBluetooth

2.1 Pod File Content

```
source 'https://github.com/CocoaPods/Specs.git'

platform :ios, '9.0'

use_frameworks!

workspace 'YourWorkSpace'

target 'YourTargets' do

xcodeproj 'YourProject.xcodeproj'

pod 'AFNetworking', '~> 3.0'

pod 'Alamofire', '~> 4.3'

end
```

二、開始使用 SDK

1.初始化 BeaconAPI

```
_ = BeaconAPI.sharedInstance.initAPI(APPID: AIRYZON_APP_ID, APPKey: AIRYZON_API_KEY)
```

2.申請使用者

```
public func RegisterUser( UserID userId:String, completeHandler handler:ApiCompletionHandler?) -> Void
```

UserID：使用者帳號

建立 ApiCompletionHandler 實例

```
BeaconAPI.sharedInstance.RegisterUser(UserID: userID, completeHandler: { (response) -> Void in
        if let dic:[String:Any] = response
        {
            let status:String = dic["status"] as! String
            let msg:String = dic["msg"] as! String
            if status == "OK"
            {
                //註冊成功
            }
            else
            {
               //註冊失敗
            }
        }
    })
```

response 格式範例：

```
{
 "status": "OK",
 "msg ": ""
}
```

status：值”OK”為申請成功，”Fail”為失敗

msg：若 status 為”OK”，則為空值，反之，會回傳錯誤訊息

3.查詢使用者裝置清單

```
public func getUserDeviceList( UserID userId:String, completeHandler handler:ApiCompletionHandler?) -> Void
```

UserID：使用者帳號

response 格式範例：

```
status：值”OK”為查詢成功
msg：回傳訊息
list    ：裝置清單
```

4.搜尋裝置

```
public func scanBeacon(Timer timer:TimeInterval,_ completionHandler: @escaping ScanCompletionHandler ) -> Void
```

Timer：需要 Scan 的秒數

建立 ScanCompletionHandler 實例

```
BeaconAPI.sharedInstance.scanBeacon(Timer: 2.0, { (state:BeaconAPI.ScanState) -> Void in

            if state == BeaconAPI.ScanState.Success

            {

                    // var discoveredPeripherals:[mBeacon]

              var discoveredPeripherals:[mBeacon] =

                                                BeaconAPI.sharedInstance.getBeaconList()

            }

            else if state == BeaconAPI.ScanState.Scanning

            {
```

```
            print("still scanning")
        }
    })
```

mBeacon 類別

beaconDeviceName()：裝置名稱

macId：唯一識別碼，Beacon 硬體編號

5.配對裝置

```
public func bindBeaconWithUser( UserID userId:String, BeaconID beaconId:String, NickName name:String,
completeHandler handler:ApiCompletionHandler?) -> Void
```

UserID：使用者帳號

BeaconID：Beacon 硬體編號

NickName：裝置暱稱

response 格式範例：

status：值”OK”為綁定成功

msg：回傳訊息

6. 解除配對裝置

```
public func unBindBeaconWithUser( UserID userId:String, BeaconID beaconId:String, completeHandler
handler:ApiCompletionHandler?) -> Void
```

UserID：使用者帳號

BeaconID：Beacon 硬體編號

response 格式範例：

```
status：值”OK”為解除綁定成功
msg：回傳訊息
```

7.開通裝置

```
public func activateBeacon(_ beacon:mBeacon?, _ completionHandler: @escaping ActivateCompletionHandler) -> Void
```

建立 ActivateCompletionHandler 實例

```
BeaconAPI.sharedInstance.activateBeacon(beacon, {(state:BeaconAPI.ActivateState) -> Void in
            if state == BeaconAPI.ActivateState.Success
            {
                //開通成功
            }
            else
            {
                //開通失敗
            }
})
```