

# WiFi Audio SDK for iOS

Version 7.1.0

## SDK Guide

### Contents

<b>iOS settings</b>	<b>2</b>
<b>Permissions requested</b>	<b>2</b>
<b>Initial setup of a new WiFi Audio app in Xcode</b>	<b>3</b>
Import the WiFi Audio SDK	3
<b>Additional topics related to the usage of the Example app</b>	<b>6</b>
Special configuration for your own Firebase project	6
<b>Setup for ExampleApp in Xcode</b>	<b>8</b>
<b>Exploring the sample application</b>	<b>9</b>
1) Connection methods:	9
2) Start and stop Audio	9
3) Use of beacons	9
4) Disconnect	10

# iOS settings

Min SDK version	11
Target SDK version	11
Compile SDK version	17

## Permissions requested

- Background mode
- Audio
- AirPlay
- Picture in Picture
- Local network usage (Required since iOS 14)
- Location (background and foreground)
- Camera usage
- Tracking usage

ExampleApp

ExampleApp

ExampleApp

General

Signing & Capabilities

Resource Tags

Info

Build Settings

Build Phases

Build Rules

PROJECT

ExampleApp

TARGETS

ExampleApp

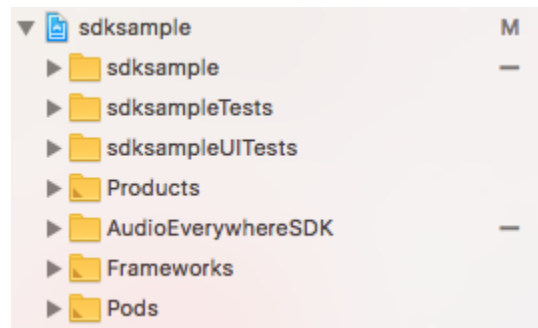
Custom iOS Target Properties

Key	Type	Value
> Required device capabilities	Array	(1 item)
Privacy - Tracking Usage Description	String	Needed for beacons use
InfoDictionary version	String	6.0
> Supported interface orientations (iPad)	Array	(4 items)
Requires Full Screen	Boolean	YES
Appearance	String	Light
Bundle name	String	ExampleApp
View controller-based status bar appearance	Boolean	NO
Main storyboard file base name (iPad)	String	AEAStoryboard_iPad
Status bar style	String	Transparent black style (deprecated)
Privacy - Location Always and When In Use Usage Des...	String	Needed for beacons use
Application requires iPhone environment	Boolean	YES
Application uses Wi-Fi	Boolean	YES
Bundle display name	String	ExampleApp
Privacy - Location When In Use Usage Description	String	Needed for beacons use
> Required background modes	Array	(2 items)
Privacy - Camera Usage Description	String	scan QR codes
Bundle version string (short)	String	\$(MARKETING_VERSION)
Privacy - Location Always Usage Description	String	Needed for beacons use
> Supported interface orientations	Array	(1 item)
Application supports iTunes file sharing	Boolean	NO
Bundle OS Type code	String	APPL
Default localization	String	en
> Bonjour services	Array	(2 items)
Bundle version	String	\$(CURRENT_PROJECT_VERSION)
> Fonts provided by application	Array	(3 items)
Status bar is initially hidden	Boolean	NO
Launch screen interface file base name	String	LELaunchScreen
Supports opening documents in place	Boolean	NO
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
> App Transport Security Settings	Dictionary	(1 item)
Executable file	String	\$(EXECUTABLE_NAME)
Main storyboard file base name	String	AEAStoryboard_iPhone
Privacy - Local Network Usage Description	String	This app will be able to discover and connect to devices on the local network.
App Uses Non-Exempt Encryption	Boolean	NO

# Initial setup of a new WiFi Audio app in Xcode

## Import the WiFi Audio SDK

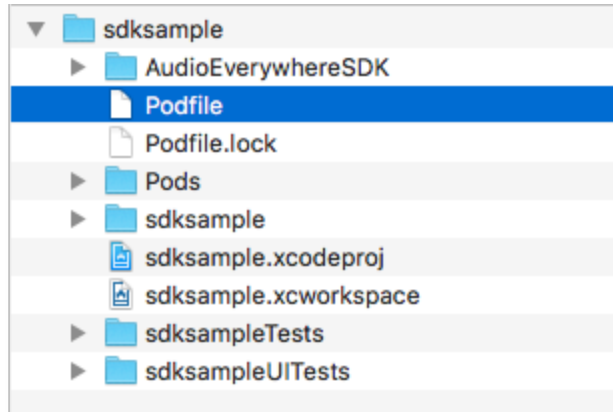
1. Download SDK using your credentials.
2. Unzip downloaded files on your computer (any directory).
3. Copy the folder directly into xcode named “AudioEverywhereSDK” into your project under `/${projectName}` following this structure:



A prompt window will appear, select “Copy items if needed”



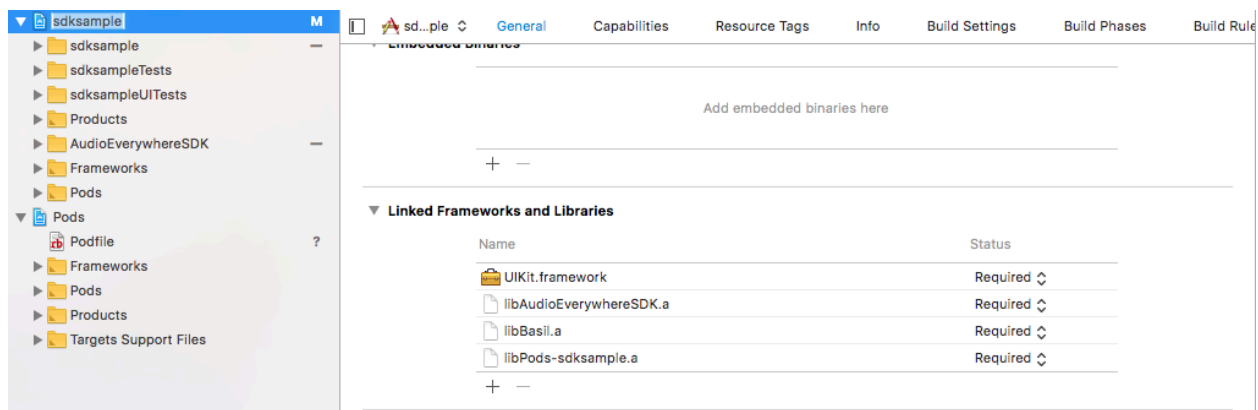
4. Add the PodFile (or modify if is already created)
  - a. Create a file named “PodFile” (only if it doesn’t exist) on the root of your project (same level as “.xcodeproj” file)



- b.** Open this file (**IMPORTANT**)
- c.** Copy (or add) the content from the example app provided.
- d.** Open a terminal
- e.** Go to your project's directory (where you created the file on step 4.a)
- f.** Run command "pod install" (If you don't have CocoaPods installed please visit <https://cocoapods.org/> to get instructions on how to configure it)

## 5. Add UIKit Framework (If you're not using it already)

- a.** Click on your project
- b.** Click on General tab
- c.** Find "Linked Framework and Libraries" section
- d.** Click on the plus sign
- e.** Look for UIKit.Framework
- f.** Add it to your project

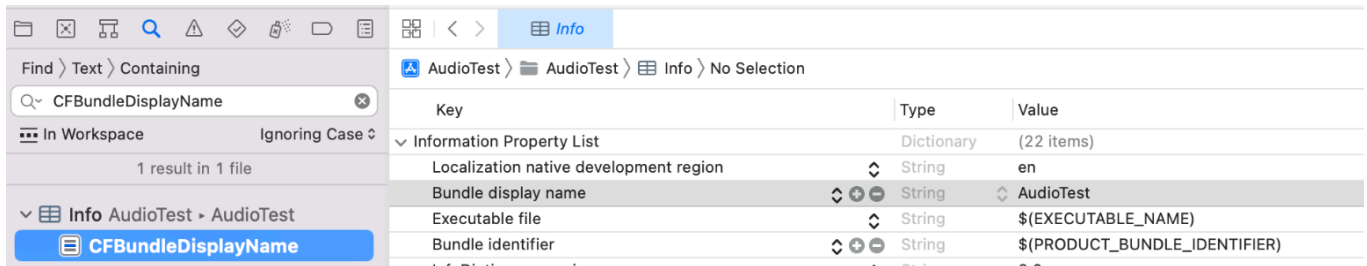


## 6. Set up CFBundleDisplayName: (**IMPORTANT**)

**a.** Open the Info.plist file

**b.** Look for the CFBundleDisplayName property (Bundle display name)

**c.** Add a String value to identify the app (Mainly for audio information)



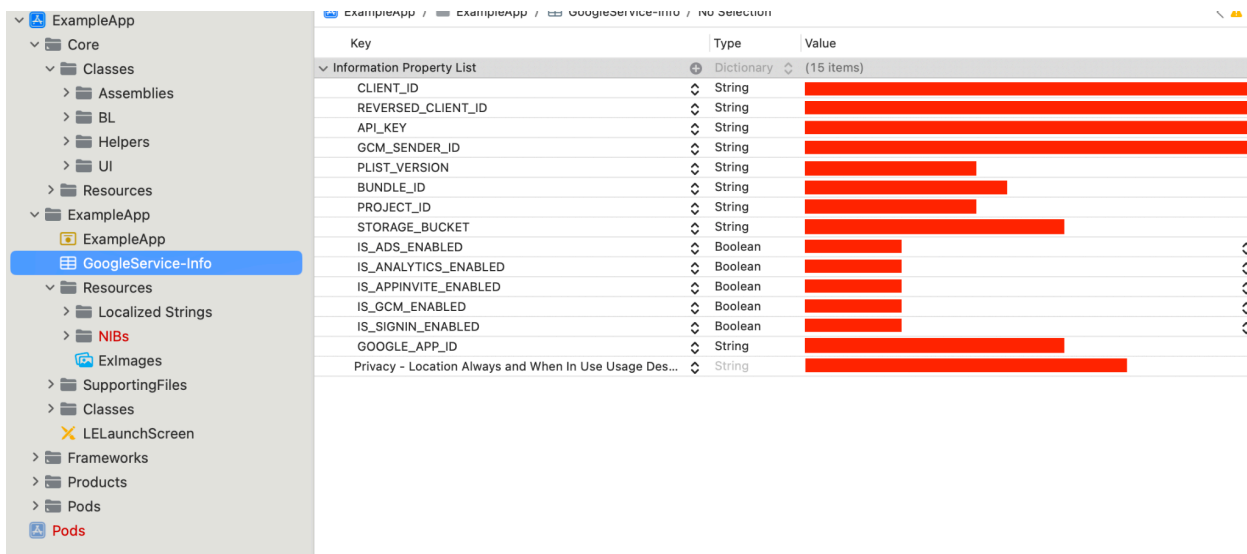
# Additional topics related to the usage of the Example app

Currently, the main application ListenWiFi scans and opens dynamic links generated for the Listen Technologies environment. For the usage of another company, a new Firebase project to support and generate the dynamic links functionality should exist and link it to the example app.

**Important note:** The Dynamic Link functionality is now deprecated but if you have a previous configuration set, it can be used until August 2025. [More information](#)

## Special configuration for your own Firebase project

1. Check the information in the file GoogleService-Info and change it with your information.

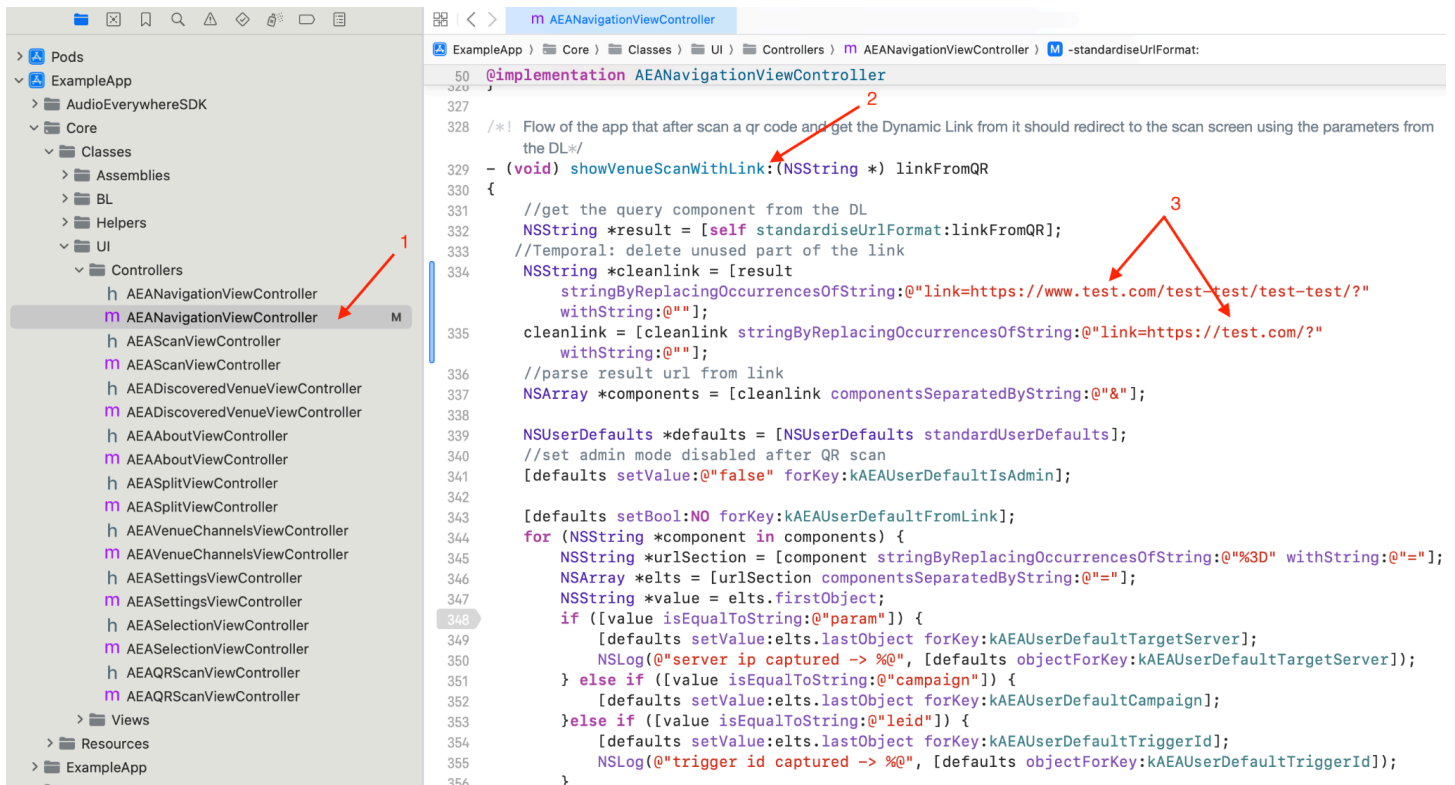


2. Once the replacement of the firebase project information is done, you should also replace the information in the following method to correctly handle the Dynamic Link.

2.1. Open AENavigationViewController

2.2. Identify the method showVenueScanWithLink

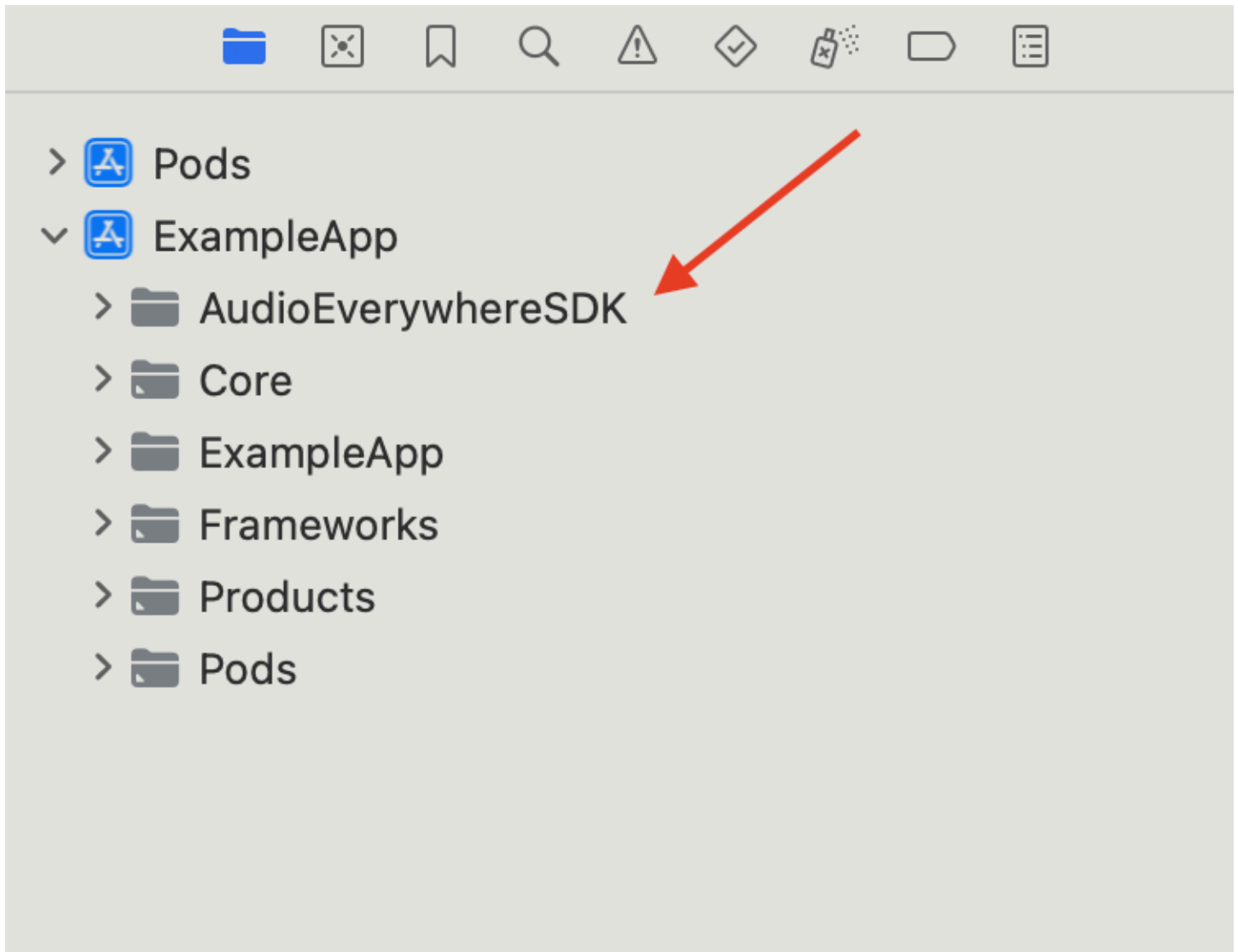
2.3. Replace the current link for your own link.



# Setup for ExampleApp in Xcode

Recommended: Use XCode version 15.4 (15F31d)

1. Replace the folder with the SDK package with the latest version of the SDK.
2. Review and replace the Pod file with [this](#) information if it's needed.
3. Install the pods for the example app.



# Exploring the sample application

## 1) Connection methods:

There are multiple ways to connect to a server and this method has an automated execution to try every method if another is not successful.

The methods available are:

- mDNS
- Remembered IP (from a previous connection)
- Scan of QR code with a Dynamic Link
- Local network discovery
- Manual connection with IP or hostname.

- (void) scanForAudioEverywhereService

## 2) Start and stop Audio

You can start and stop the playback through the connection manager. Check the simple methods that will do the handling for you.

- (void) startPlaybackOnChannelWithIndexPath:(NSIndexPath \*) indexPath

- (void) stopPlaybackOnActiveChannel

## 3) Use of beacons

This method will start all the needed components to detect and react to the beacons functionality

```
//setup location for monitor beacons  
[self.audioEverywhereManager setupLocationManager];
```

## 4) Disconnect

Disconnection method is used for a manual disconnection from the user or algo to handle errors or handle some environment circumstances like when there are no servers available on the current network.

Handle audio error:

- (void) updateDisconnectCountdown

Manual disconnection:

- (IBAction)disconnect:(UIButton \*)sender